PATH PLANNING FOR AN ARTICULATED TRANSPORTER/MANIPULATOR SYSTEM

BY

SHIH-CHIEN CHIANG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1992

ACKNOWLEDGEMENTS

The author would like to express his appreciation to his supervisory committee chairman, Dr. Joseph Duffy, for his support of this research. Sincere gratitude is expressed to his supervisory committee cochairman, Dr. Carl D. Crane III, for his invaluable inspiration, guidance and enthusiasm throughout the research. Special thanks are due to his graduate committee members, Dr. Roy Harrell, Dr. Ali Seireg, and Dr. John Schueller for their comments on this work. He would like to thank his colleagues in the Center for Intelligent Machines and Robotics (CIMAR) for sharing their knowledge.

Finally, the author extends his deepest appreciation to his parents for their moral support and to his beloved wife, Grace, for her encouragement, understanding and support.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

PATH PLANNING FOR AN
ARTICULATED TRANSPORTER/MANIPULATOR SYSTEM

By

Shih-chien Chiang

August 1992

Chairman: Dr. Joseph Duffy
Major Department: Mechanical Engineering

This work addresses the three-dimensional path planning for an Articulated Transporter/Manipulator System (ATMS) in a given working environment. The working environment consists of horizontal planes at different elevations. Movement between planes and over obstacles is the unique design feature of the ATMS. A vertical motion capability provides the ATMS a new ability which can be used to advantage in the generation of collision free paths. It also complicates the path planning process, however, by not being constrained to a two-dimensional environment.

A hierarchical structure of path planning is developed to decompose the three-dimensional path planning problem into several two-dimensional sub-problem. In the first level of planning, the sequence of intermediate planes (via planes)

between starting and goal points is planned. This reduces the path planning to that of navigating on each via plane. In the second level of planning, buffer zones and pseudo-obstacles are designed to represent the Jump-Over (J-O) obstacles. This special representation of J-O obstacle simplifies the path planning to a two-dimensional case. In the third level, modifications are made to ensure the planned path is suitable for the ATMS to follow.

In this research, an obstacle may not always be treated as an obstacle and free space may not always be treated as free space either. *"An obstacle, not an obstacle; free space, not free space"* is the art of the path planning of the ATMS. Intelligence is implanted in the system to recognize the situation and plans the collision free path.

The main emphasis of the study is on finding a feasible path based on geometric considerations with the assumption that the system actuators are capable of executing the necessary movements.

The path planning algorithms are successfully implemented on the Silicon Graphics 4D-310VGX workstation. Motions of the ATMS are animated in a designed working environment which consists of more than 40 obstacles to verify the results. The computation time varies from less than 1 second to approximate 3 seconds depending on the specified start and goal points.

CHAPTER 1
INTRODUCTION

An Articulated Transporter/Manipulator System (ATMS) is being designed for use in nuclear power stations. A key design specification of this system is that it be able to maneuver through a complex obstacle strewn environment. The environment consists of several planes which are in different elevations. Elaborate pipe and cable networks, dikes, and stairways are found on the plane. It is assumed that only one access way exists between two planes. The ability to move throughout this type of environment will allow the robotic system to perform inspection, maintenance, and emergency response tasks in high radiation areas.

The ATMS design which is analyzed in this dissertation is comprised of eighteen individual segments which provide both maneuverability and locomotion.[1] Each segment is 11.5 inches wide, 13 inches high, and 24 inches in length, and has a pair of motor driven wheels to provide traction for forward and

---

[1] A modification to the ATMS concept was made in 1992 whereby each segment could change in length from 24 to 40 inches [1]. Most of the analysis in this dissertation can be applied directly to the new design concept.

backward motion. Segments are connected in series by a pitch joint and a yaw joint (see Figure 1.1).

In addition to the horizontal motion (see Figure 1.2), a significant feature of the design is that the ATMS will be able to cross over horizontal gaps of up to twelve feet in length. This design also enables the ATMS to cross over obstacles by 'bridging' through the air. The capability of vertical motion of the ATMS (see Figure 1.3) makes it unique from the mobile vehicles which travel in two-dimensional space only.

The analysis for vertical and horizontal motions of the ATMS has been completed [2,3]. This research is focused on autonomous path planning for the ATMS in a complex obstacle strewn environment. Vertical motion capability of the ATMS provides it with better maneuverability in avoiding collisions with obstacles. It also makes path planning more complicated than for traditional mobile vehicles because the planning is not confined in a two-dimensional space. Part of the planned collision-free path segments will be located on horizontal planes in different levels and part of the path segments will be in vertical planes.

The unique vertical motion capability of the ATMS makes the path planning of the ATMS no longer a two-dimensional problem. The objective of this research is to develop a strategy to break down the problem to several sub-problems and use a two-dimensional planning algorithm to plan a three-

dimensional collision-free path for the ATMS in a given working environment.

The problem can be stated as follows:

Given: 1. Start and goal position and orientation of the lead segment of the ATMS on certain planes.

2. The information of the working environment.

Find: A "good" collision-free path between the start and goal points.

To fully utilize the ATMS's horizontal and vertical motion capabilities, sufficient information about obstacles (quantitative and descriptive) must be provided to the system for decision-making during path planning. In Chapter 3, three types of obstacle are defined to provide descriptive information about obstacles in the working environment. To reduce path planning to a two dimensional problem, a special means of modelling the working environment is essential. The method used to model the working environment is also described in Chapter 3. A hierarchical structure of path planning is introduced in Chapter 4. The path planning problem is solved by breaking it down into two related sub-problems. The algorithms have been successfully implemented on a Silicon Graphics workstation, the simulation results are displayed in Chapter 5.

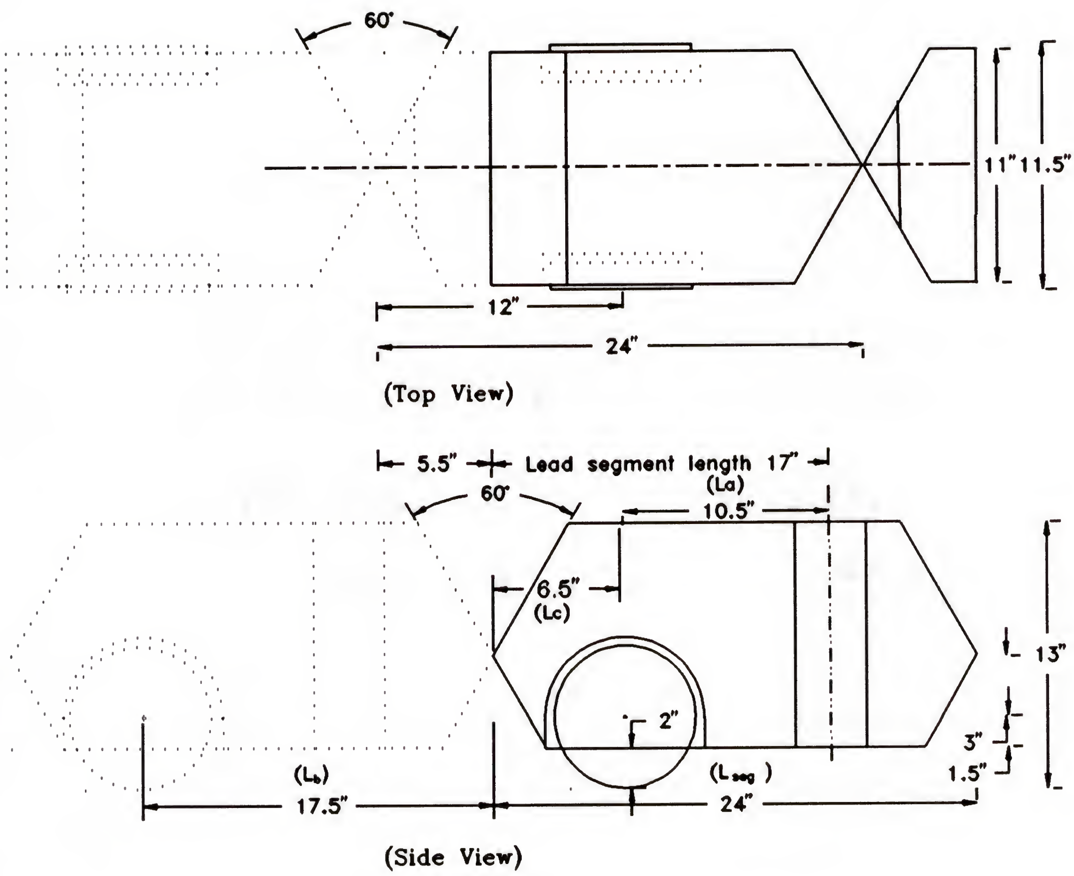


Figure 1.1 Dimensions of ATMS

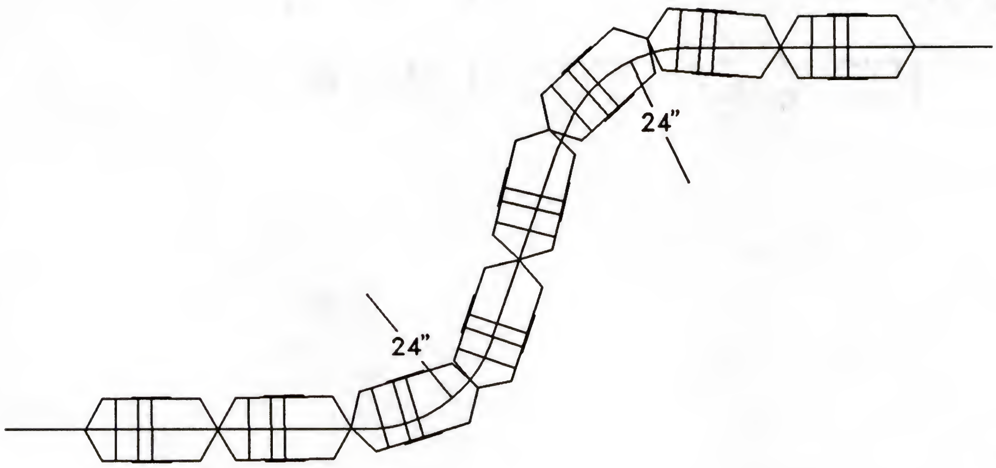
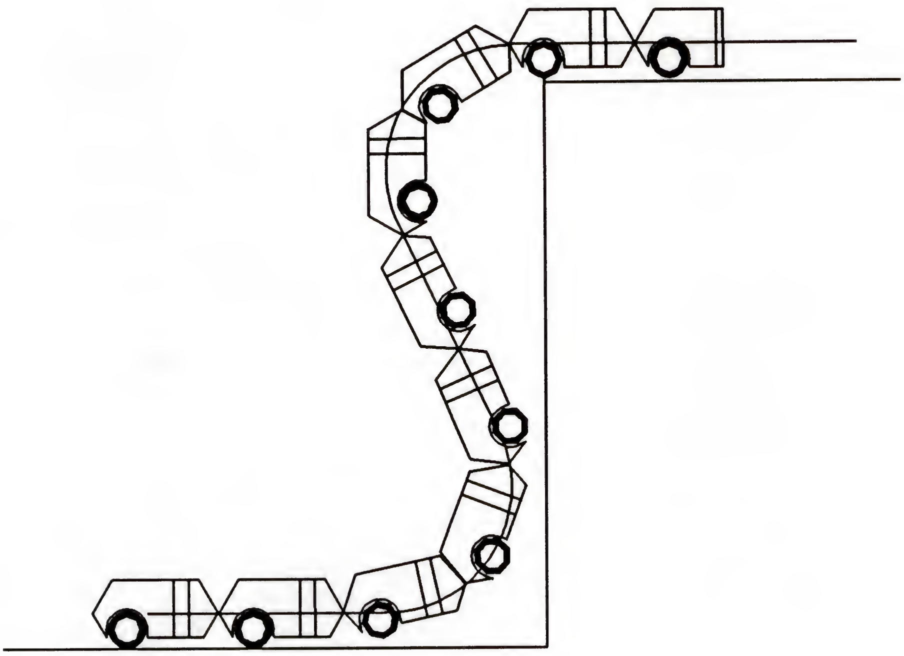


Figure 1.2 Horizontal Motion (Top View)

Vertical Motion (Moving to Higher Level)



Vertical Motion (Jump Over Pipe)

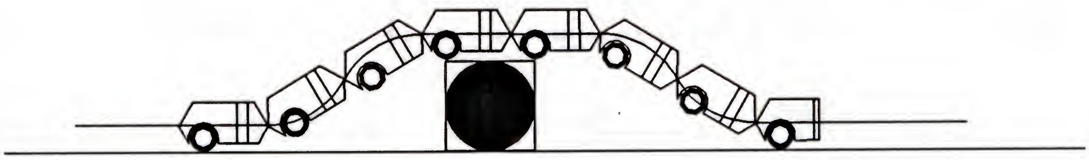


Figure 1.3 Vertical Motion (Side View)

CHAPTER 2
REVIEW OF LITERATURE

2.1 Mobile Robots for Nuclear Power Plant

After the Chernobyl disaster there has been increased concern about the safety of the world's nuclear plants and the role of human factors in nuclear plant instrumentation and control and the use of robotics technology to replace workers. The U.S. Three Mile Island nuclear accident has already prompted new developments in the use of robotics in high radiation areas so the occupational radiation exposures can be kept as low as reasonably achievable.

The mobile robots which are under development for use in nuclear plants can be classified into three basic types: (1) wheels and track type, (2) legs type, and (3) articulated body type. The wheels and track type is the most common design for the mobile robot due to its easy locomotion control and good payload capacity. Examples of this type of mobile robot are: SURBOT [4,5], SURBOT-T [6,7], SURVEYOR [8,9], SIMON [10], HERMIES-III [11], AMR [12,13], and TBIS[14]. The legs type of mobile robot has high terrain adaptability, but the payload capability is restricted and the control is more complicated. The ODEX (or ROBIN) [15,16] is an example of this type of

mobile robot. There are also some mobile robots which are designed by combining wheels and track and legs, such as MF3 [17], MRV-1 [18], and C/V ROBOT [19,20]. The articulated body type of mobile robot has excellent terrain adaptability, payload capability and high mobility. KR-I [21] is an example of this type of mobile robot.

For a mobile robot that can cross over a horizontal gap of up to twelve feet in length and work in a environment made up of elaborate pipe and cable networks, dikes, and stairways, the articulated body type is believed to be the best choice.

2.2 Path Planning

Path planning for mobile robots has been a subject of considerable interest for AI and robotics researchers. Many techniques that compute paths for a land-based robot assuming a two dimensional model of the world have been developed. The problem of path planning for mobile robots is to find a collision-free, cost-effective path between a start and a goal position in a clustered environment. In an entirely known environment, the find-path problem is usually solved by a graph-search algorithm. The graph can be a visibility graph[22], a generalized Voronoi graph [23] or a designation of free convex spaces [24]. Other methods also exist for generating a graph representing the topology of the free region in space.

Udupa [25] was one of the first to extensively use the concept of the configuration space (Cspace) to develop an obstacle avoidance algorithm and allow treating the moving object as a point. The method used to find a path was not concerned with the optimality of the path but iteratively tried different alternatives until a safe path was found.

Lozano-Perez [22,26,27] used Cspace to plan collision-free motions. Alternative strategies were used for finding safe paths. One approach was establishing the visibility graph (Vgraph) of the Cspace obstacles and used graph searching technique to find the shortest path between the start and the goal among obstacles. The approach was very susceptible to model inaccuracy and position error. Another approach was to conduct the path search using a representation of the complement of the Cspace obstacles, called free space. A free space graph (FSG) was built and the shortest path was searched in the free space graph from the cell including the start to that including the goal.

Andresen et al. [28] developed a system of algorithms for mobile robot path planning based on a multiresolution representation of the robot's immediate environment. The multiresolution representation used is the quadtree. Large areas of free space (or of obstacles) can be represented by a few large blocks in the quadtree and can be dealt with as units by the planning algorithms. A path generated from a quadtree is a sequence of blocks through which it is possible

for the robot to move. The quadtree always has a great number of nodes, and hence it is not trivial to select the collision-free path out of the quadtree in economical calculation time. To accomplish this, Noborio et al. [29] introduced a path graph built as small as possible on the quadtree while investigating a reasonable collision-free path.

Khatib [30] presented a low-level, real-time obstacle avoidance approach based on the artificial potential field concept. The movement of the manipulator or mobile robot is governed by the attractive and repulsive forces in the artificial potential field. This approach reduced the high level path planning burden by extending low level control capability. At high level control, instead of finding an accurate collision-free path, it only generated a global strategy for the manipulator's path in terms of intermediate goals. The detailed geometry and motion of the manipulator and obstacles were considered at low level to produce appropriate commands to attain each of these intermediate goals.

Hague et al [31] presented a simple algorithm, based on the artificial potential field, capable of planning translational motions for a convex polygonal object amongst polygonal obstacles. The potential at any particular point is defined as being the maximum of the potential due to each obstacle, plus a component decreasing linearly toward the goal location. Instead of actually computing the potential, the

algorithm only calculated the resultant force acting on the moving object due to the potential field. A heuristic based on the concept of taking moments allowed the algorithm to be extended to include a rotational degree of freedom.

Takahashi and Schilling [23] presented a generalized Voronoi diagram (GVD) algorithm for planning in a planar workspace populated with polygonal obstacles. For a polygonal workspace populated with polygonal obstacles, the GVD consists of a network of linear and parabolic line segments. Once the GVD representation of free space is obtained, the shortest path on the GVD having an adequate radius is easily found using graph theory techniques. Four heuristic techniques were employed to align a rectangle along the GVD path. The simplest one being to align the major axis of the moving rectangle with the tangent to the Voronoi diagram. Further heuristics are tried in turn if previous attempts have not yielded a collision free path.

Kuan, Zamiska and Brooks [32] presented an algorithm that decomposed the free space into nonoverlapping geometrically shaped primitives: generalized cones (channels) and convex polygons (passage regions). The passage regions are connected by channels. The two entrance points for each channel and the critical points around the passage regions built up the searchable graph. The A^* search algorithm [33] is used to find the optimal path in the searchable graph.

Arkin [34] used a hybrid vertex-graph free-space representation for global path planning. The hybrid vertex-graph free-space representation (meadow map) based upon the decomposition of free space into convex regions. This "meadow map" is produced via the recursive decomposition of the initial bounding traversable area and its associated modeled obstacles. A major advantage of the meadow map representation is the ease with which representation of objects, landmarks, terrain features, etc., can be expanded.

Borenstein and Koren [35] developed a real-time obstacle avoidance method, Vector Field Histogram (VFH), which permits the detection of unknown obstacles and avoids collisions while simultaneously steering the mobile robot toward the target. The principles of the VFH method are

a. Continuously up-date the Histogram Grid with range data sampled from the onboard range sensor.
b. Construct the one dimensional Polar Histogram around the momentary location.
c. Sector with low obstacle density in the Polar Histogram and close to the target direction is selected as the robot's steering direction.

The VFH algorithm has been tested on an experimental mobile robot, with a ring of 24 ultrasonic sensors, traversing a densely cluttered obstacle course.

The path planning algorithms stated above treat all the obstacles in an uniform manner, i.e. path does not intersect

with obstacle in the two-dimensional space. However, for the case of path planning for the ATMS, the planned path is allowed to intersect with certain obstacles in the two-dimensional space in order to plan a vertical path over the obstacle. A new strategy of path planning has to be developed so it can plan a horizontal path by avoiding intersection with obstacles and plan a proposed vertical path which intersects with the obstacle in the two-dimensional space. The proposed vertical path is modified to a real vertical path in the third dimension afterward.

CHAPTER 3
MODELLING WORKING ENVIRONMENT

An objective of properly modelling a working environment is to reduce the computational efforts required for path planning. In this chapter, the definitions of types of obstacles which modelled are described first. Second, an obstacle expansion algorithm is presented. In the obstacle expanded environment the lead segment of the ATMS can be treated as a point, and the path planning on a horizontal plane is simplified to a two-dimensional graph search. Third, a special representation for the obstacle which the ATMS can jump over is shown. This representation allows the proposed vertical path to be planned using the same process utilized in the planning of the horizontal path.

3.1 Obstacles in the ATMS Working Environment

For the case that the ATMS travels on a horizontal plane, like a mobile vehicle which travels in two dimensional space, the only motion needed for the ATMS is horizontal motion. The only way that the ATMS can avoid colliding with obstacles is by going around them. There are also cases in

which horizontal motion is not the best choice to pass the obstacle. Vertical motion, like jump over or land on, may be more efficient for passing the obstacle. Whether a vertical motion is to be planned to pass the obstacle depends on: 1) the characteristics of the obstacle, 2) the vertical motion constraints, and 3) the current location of the ATMS relative to the obstacle. Information about the characteristics of the obstacles in the working environment must be established in the database of the system so that the ATMS can make a proper decision based on this information.

Three obstacle types are defined based on the characteristics of the obstacle:

I. **Go-Around obstacle (G-A obstacle)**

An obstacle which the collision free path for the ATMS can only be planned by going around it.

II. **Jump-Over obstacle (J-O obstacle)**

An obstacle which is long, narrow and low that the ATMS can jump over. A pipe on the floor and a dike are examples of this type of obstacle.

III. **Land-On obstacle (L-O obstacle)**

An obstacle which is large and the ATMS is allowed to travel on the top of the obstacle.

Through these defined obstacles, the ATMS will be better informed about the obstacle that it encounters. For example, a pipe laid on the ground can be jumped over by the ATMS. The pipe will be described as a J-O obstacle to indicate that

vertical motion can be adopted to pass it. During path planning, the system will not only treat it as a G-A obstacle and try to plan a horizontal path to go around it but also recognize it as a J-O obstacle and try to plan an alternative vertical path to jump over it. From the example, it is clear that defining an obstacle as a J-O type or L-O type obstacle reveals the characteristics of the obstacle. The definitions are not used to uniquely classify the obstacle.

3.2 Obstacle Expansion

The concept of configuration space is adopted in modelling the working environment. In the obstacle expanded environment, the lead segment of the ATMS can be treated as a point. The path planning for the ATMS becomes planning a path for a point to travel in the obstacle expanded environment.

In many situations, the movement of the ATMS is on a horizontal plane along the edges of an obstacle and turns around the corner of the obstacle. For the case that the ATMS moves along the edge of the obstacle, the minimum amount of required expansion of the obstacle is half the width of the ATMS segment. However, uniform expansion of edges of the obstacle by that amount is not feasible for the case in which the ATMS turns around the corner of the obstacle due to the requirement of the minimum turning radius for the ATMS. In order to expand the obstacle properly, primary concern must be

given to the expansion of the corner of the obstacle. The amount of expansion on each edge of the obstacle depends on the expansion of the corners of the obstacle. In this research, an obstacle on the horizontal plane is assumed to be a convex polygonal or can be decomposed into convex polygons.

3.2.1 Minimum Amount of Expansion

The first task of the obstacle expansion is to determine the required minimum amount of expansion. For the ATMS, a Follow-the-Leader algorithm is used in the horizontal motion. The odd segments are always positioned on the horizontal path and the even segments are positioned according to the adjacent odd segments (see Figure 1.2). Expanding an obstacle by the amount of half the width of the ATMS segment is enough while the ATMS is moving along the edge of the obstacle. But more space is required while the ATMS is moving on a circular path around the corner of the obstacle . The amount of expansion needed for the odd segment of the ATMS to move on the circular path is **d1** as shown in Figure 3.1. The amount of expansion needed for the even segment of the ATMS to move on the circular path is **d** as shown in Figure 3.2. It is clear that **d** is larger than **d1**; so for the obstacles in the ATMS working environment, **d** should be considered as the minimum amount of expansion to expand the corners of the obstacle.

3.2.2 Algorithm of Obstacle Expansion

As stated previously, the primary concern of the obstacle expansion in the ATMS environment are obstacle corners. In the obstacle expansion algorithm, the corners of the obstacle are expanded first, and then the expansion of the edge is based on the expanded corners of the obstacle. The algorithm can be summarized as three major steps as follows:

1. Find the expansion circle for each of the corners of the obstacle. The expansion circle ensures a collision free path around the corner of the obstacle if a circular path is planned on the expansion circle.
2. Based on the locations of the expansion circles, find the proper common tangent line for each pair of the adjacent expansion circles. These tangent lines are the expanded edges of the obstacle.
3. Find the intersection point of each pair of adjacent tangent lines, and these points are the vertices of the expanded obstacle.

The details of each step of the obstacle expansion are described later.

3.2.2.1 Expansion of the corners of the obstacle

In this section, the method used to find the expansion circle for each of the corners of the obstacle is described. As shown in Figure 3.3, V_1, V_2, V_3, V_4, and V_5 are the vertices

of the obstacle and they are in counter clockwise order. The coordinates of the vertices of the obstacle have been stored in the database of the system. The expansion circle **C** is located on the bisector of $\angle V_1V_2V_3$ with a radius of required turning radius (R_turn) of the ATMS. The expansion circle can be located by the coordinates of the vertices, the minimum amount of expansion, and R_turn.

The bisector of $\angle V_1V_2V_3$ can be determined by finding the point I which is located on the bisector. The lines L_1 and L_2 (in Figure 3.3) are respectively parallel to and shifted to the left (as move along the edges of the obstacle in a counter clockwise direction) of the edges V_1V_2 and V_2V_3 of the obstacle by an arbitrary distance of S. The equations for lines L_1 and L_2 can be expressed as

$$L_1: \quad A_1x + B_1Y + C_1 = 0$$
$$L_2: \quad A_2x + B_2y + C_2 = 0. \qquad (3.1)$$

where the coefficients A_1, B_1, C_1, A_2, B_2, and C_2 can be calculated from the coordinates of the vertices $V_1(x_1, y_1)$, $V_2(x_2, y_2)$, and $V_3(x_3, y_3)$.

$$A_1 = y_2 - y_1$$
$$B_1 = x_1 - x_2$$
$$C_1 = -A_1x_1 - B_1y_1 + S * (A_1^2 + B_1^2)^{1/2}$$
$$A_2 = y_3 - y_2$$
$$B_2 = x_2 - x_3$$

$C_2 = -A_2x_2 - B_2y_2 + S * (A_2^2 + B_2^2)^{1/2}$.

The intersection point $I(x_I, y_I)$ of the lines L_1 and L_2 is a point located on the bisector of angle $\angle V_1V_2V_3$. The coordinates of point $I(x_I, y_I)$ can be calculated by using Cramer's rule,

$$x_I = \frac{\begin{vmatrix} B_1 & C_1 \\ B_2 & C_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}}$$

$$y_I = \frac{\begin{vmatrix} C_1 & A_1 \\ C_2 & A_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} \qquad (3.2)$$

After the coordinates of point I are found, the distance (D) between points I and V_2 can be calculated. Knowing the distance (D), turning radius (R_turn) and d (minimum amount of expansion), the parameter t which is expressed as Eq.3.3 can be decided and used to find the coordinates of the center $O(x_o, y_o)$ of expansion circle **C**.

$$t = (R\_turn - d) / D \qquad (3.3)$$

The coordinates of the center O are calculated by

$$x_o = x_2 + (x_I - x_2) * t$$
$$y_o = y_2 + (y_I - y_2) * t \tag{3.4}$$

The expansion circle ensures that the planned path is a collision free path when the ATMS moves around the corner of the obstacle.

3.2.2.2 Expansion of the edges of the obstacle

In this section, the edges of the obstacle are expanded based on the expansion circles found in the previous section. The points $O_1(x_{01}, y_{01})$, $O_2(x_{02}, y_{02})$, and $O_3(x_{03}, y_{03})$ in Figure 3.4 are the centers of the expansion circles for the corners V_1, V_2, and V_3 of the obstacle respectively. In order to have smooth horizontal movement of the ATMS at the transition point (straight path to circular path, or circular path to straight path), the proper common tangent line for each pair of the adjacent expansion circles is calculated and forms the edge of the expanded obstacle.

The common tangent lines L_3 and L_4 in Figure 3.4 can be found by shifting the lines O_1O_2 and O_2O_3 to the right (as move along the edges of the obstacle in a counter clockwise direction) by the distance of R_turn. L_3 and L_4 can be expressed as

$$L_3: \quad A_3x + B_3 + C_3 = 0$$
$$L_4: \quad A_4x + B_4 + C_4 = 0 \tag{3.5}$$

where the coefficients of the line equations are calculated as follows:

$$A_3 = y_{02} - y_{01}$$

$$B_3 = x_{01} - x_{02}$$

$$C_3 = -A_3x_{01} - B_3y_{01} - R\_turn * (A_3^2 + B_3^2)^{1/2}$$

$$A_4 = y_{03} - y_{02}$$

$$B_4 = x_{02} - x_{03}$$

$$C_4 = -A_4x_{02} - B_4y_{02} - R\_turn * (A_4^2 + B_4^2)^{1/2}.$$

Representing an obstacle by expansion circles and tangent lines can achieve the purpose of obstacle expansion. However, it is not an adequate representation for path planning, a tremendous amount of computation is required to find the intermediate points of the path. A further step is taken in obstacle expansion to simplify the representation of an expanded obstacle and save computation in the process of path planning. Instead of using the expansion circles, the vertices of expanded obstacle are calculated, and the expanded obstacle can then be represented by a polygon.

The coordinates of the vertex $V_2'(x_{V2'}, y_{V2'})$ of the expanded obstacle are calculated as follows:

$$x_{V2'} = \frac{\begin{vmatrix} B_3 & C_3 \\ B_4 & C_4 \end{vmatrix}}{\begin{vmatrix} A_3 & B_3 \\ A_4 & B_4 \end{vmatrix}}$$

$$y_{V2'} = \frac{\begin{vmatrix} C_3 & A_3 \\ C_4 & A_4 \end{vmatrix}}{\begin{vmatrix} A_3 & B_3 \\ A_4 & B_4 \end{vmatrix}} \tag{3.6}$$

Figure 3.5 shows an obstacle $V_1V_2V_3V_4V_5$ is expanded by using the described obstacle expansion algorithm. The expanded obstacle is presented by the polygon $V_1'V_2'V_3'V_4'V_5'$.

The data stored in the database of the system for the expanded obstacle are

1. the coordinates of the vertices of the expanded obstacle,
2. the coefficients of the line equations which represent the edges of the expanded obstacle.

3.2.2.3 Special considerations in obstacle expansion

There are two special situations to be considered in the obstacle expansion: 1) the edge of the obstacle is short, 2) the angle of the corner of the obstacle is small.

Figure 3.6 shows an obstacle with a short edge V_1V_2 and the border lines show the edges of the expanded obstacle which is expanded by the algorithm stated in sections 3.2.2.1 and 3.2.2.2. For the case of a horizontal path is planned for the ATMS to move along the edges of the obstacle and around corners V_1 and V_2. Circular paths $\frown{T_1T_2}$ and $\frown{T_3T_4}$ are planned

for the ATMS to turn around the corner V_1 and corner V_2. It is clear that the ATMS cannot follow the path $\overset{\frown}{T_3T_4}$ after traveling on the circular path $\overset{\frown}{T_1T_2}$. The path $\overset{\frown}{T_1T_2}$ leads the ATMS to point T_2 after turning around the corner V_1 and it is not possible for the ATMS to move from point T_2 to point T_3.

The solution for this situation is shown in Figure 3.7, instead of creating two expansion circles C_1 and C_2, only expansion circle C_1 is created. Since the point P_2 is located inside of the expansion circle C_1, it can also be treated as the expansion circle for corner V_2. The edge $V_1'V_2'$ of the expanded obstacle can be decided by the line segment tangent to the expansion circle C_1 and parallel to the line segment $\overline{V_1V_2}$.

Figure 3.8 shows the case that the obstacle has a sharp corner and the obstacle in this case is over expanded. This situation can be improved by introducing an extra edge to flatten the sharp corner of the obstacle. Figure 3.9 shows the expanded obstacle for the modified obstacle. Much of the free space is saved in this way.

For a corner of the obstacle, generally, the smaller angular value of the corner has a larger amount of over expansion. However, the amount of over expansion also depends on the angular values of the adjacent corners of the obstacle, and the length of the sides of the corner of the obstacle. Introducing an extra edge to an obstacle provides a way of

modifying the obstacle when the unreasonable expansion of the obstacle is noticed.

For the case of a triangular obstacle with one short edge, only two expansion circles are created in the process of obstacle expansion. Two of the edges of the expanded obstacle are parallel by using the described algorithm of obstacle expansion, and the expanded obstacle is not a polygon. In this case, the triangular obstacle is modified by introducing an extra edge to modify the triangular obstacle to a quadrilateral obstacle.

3.3 Representation of a J-O Obstacle

The existence of a J-O obstacle on the horizontal plane causes the path planning for the ATMS to not be constrained to two-dimensional space. In order to plan the proposed vertical path in the process of horizontal path planning, a special representation of the J-O obstacle is introduced. The J-O obstacle is assumed to have a rectangular projection on the plane, or the projection on the plane can be bounded by a rectangle.

To plan a successful vertical path over a J-O obstacle, there must be an adequate distance on both sides of the J-O obstacle which is free of any other obstacle to permit the ATMS to "take-off" and "land". A buffer zone and pseudo obstacle representation of the J-O obstacle is designed to

ensure the path, which is planned in horizontal planning, over the J-O obstacle is adequate for the vertical motion of the ATMS.

3.3.1 Buffer Zone

To ensure the path planned over the J-O obstacle in the horizontal path planning satisfies the requirement of adequate distance for the ATMS to take off and land, buffer zones are created on both sides of the J-O obstacle (see Figure 3.10). The width of the buffer zones is equal to the length of the expanded J-O obstacle and the depth of the buffer zones is equal to the distance **d_buffer** which is the distance needed for the ATMS to take-off and land. In Figure 3.10, the edges B_1B_2 and B_3B_4 can be found by shifting lines J_1J_2 and J_3J_4 by the distance of **d_buffer**. Areas $B_1B_2J_2J_1$ and $J_4J_3B_3B_4$ are the buffer zones for the J-O obstacle $J_1J_2J_3J_4$.

For simplicity, in the database of the system, two buffer zones are combined into one buffer zone $B_1B_2B_3B_4$. The coordinates of the vertices of the buffer zone can be calculated by finding the intersecting points of the adjacent pair of lines. The data stored in the database for the buffer zone are the coordinates of vertices B_1, B_2, B_3, B_4, and the coefficients of the line equations which represent the edges of the buffer zone.

If a proposed vertical path is started from one side (B_1B_2 or B_3B_4) of the buffer zone and ended on the other side (B_3B_4 or B_1B_2), the path is adequate for vertical motion of the ATMS as far as the take-off and landing distances are concerned.

3.3.2 Pseudo Obstacles

The buffer zone ensures the proposed vertical path satisfies the take-off and landing distance requirement for a vertical path planned over a J-O obstacle. For the proposed vertical path to be a feasible one, it should not intersect with any other obstacle. For the case shown in Figure 3.11, there are two expanded obstacles **A** and **B** which overlap the buffer zone. If the start point of the proposed vertical path is located between P_{1a} and P_{4a} or between P_{2b} and P_{3b}, the vertical path will not have a clear landing on the other side of the J-O obstacle (for simplicity, the vertical path will be in a vertical plane which is perpendicular to line J_1J_2). Although the feasibility of the proposed vertical path can be checked after the path has been planned, it would be better not to plan an unfeasible proposed vertical path. The concept of pseudo-obstacle is introduced for that purpose.

In Figure 3.11, a feasible proposed vertical path should not be planned inside the areas $P_{1a}P_{2a}P_{3a}P_{4a}$ and $P_{1b}P_{2b}P_{3b}P_{4b}$ in the horizontal planning. These two areas should be treated as obstacles while the proposed vertical path is to be planned.

Since the obstacle is not a real obstacle, it is called a pseudo-obstacle.

The size of the pseudo obstacle is decided by the maximum width of the expanded obstacle as it overlaps the buffer zone. In Figure 3.11, the size of the pseudo-obstacle created by the expanded obstacle **A** is decided by the vertices of the expanded obstacle A which are located inside of the buffer zone. However, the size of the pseudo-obstacle created by the expanded obstacle **B** is decided by the intersection points of the obstacle with the boundary of the buffer zone.

Figure 3.12 shows the diagram of creating pseudo obstacles for a J-O obstacle. The procedures are described as follows:

1. Find the obstacles which overlap with the buffer zone.
2. Find the intersecting points of the overlapped obstacle with the boundary of the buffer zone.
3. Find the vertices of the overlapped obstacle which are located inside of the buffer zone.
4. Choose a line which is perpendicular to J-O obstacle (e.g., B_1B_4 in Figure 3.11) as a reference line.
5. Calculated the distances from points found in 3 and 4 to the reference line.
6. Choose the maximum distance point and minimum distance point to form the sides of the pseudo-obstacle.
7. Calculate the coordinates of the vertices of the pseudo-obstacle.

8. Calculate the coefficients of the line equations which represent the edges of the pseudo obstacle.

Like other types of obstacle, the data stored in the database of the system for a pseudo-obstacle are the coordinates of the vertices and the coefficients of the line equations which represent the edges of the pseudo-obstacle.

In the process of horizontal planning for passing a J-O obstacle, both possible paths (horizontal and vertical) are considered. Representing a J-O obstacle by an expanded obstacle (as described in section 3.2) allows the system to plan a horizontal path. However, representing it by buffer zone and pseudo-obstacle allows the system to plan a proposed vertical path. Fitting the buffer zone and pseudo-obstacle representations into the horizontal planning algorithm and the planning a proposed vertical path will be discussed in Section 4.2.

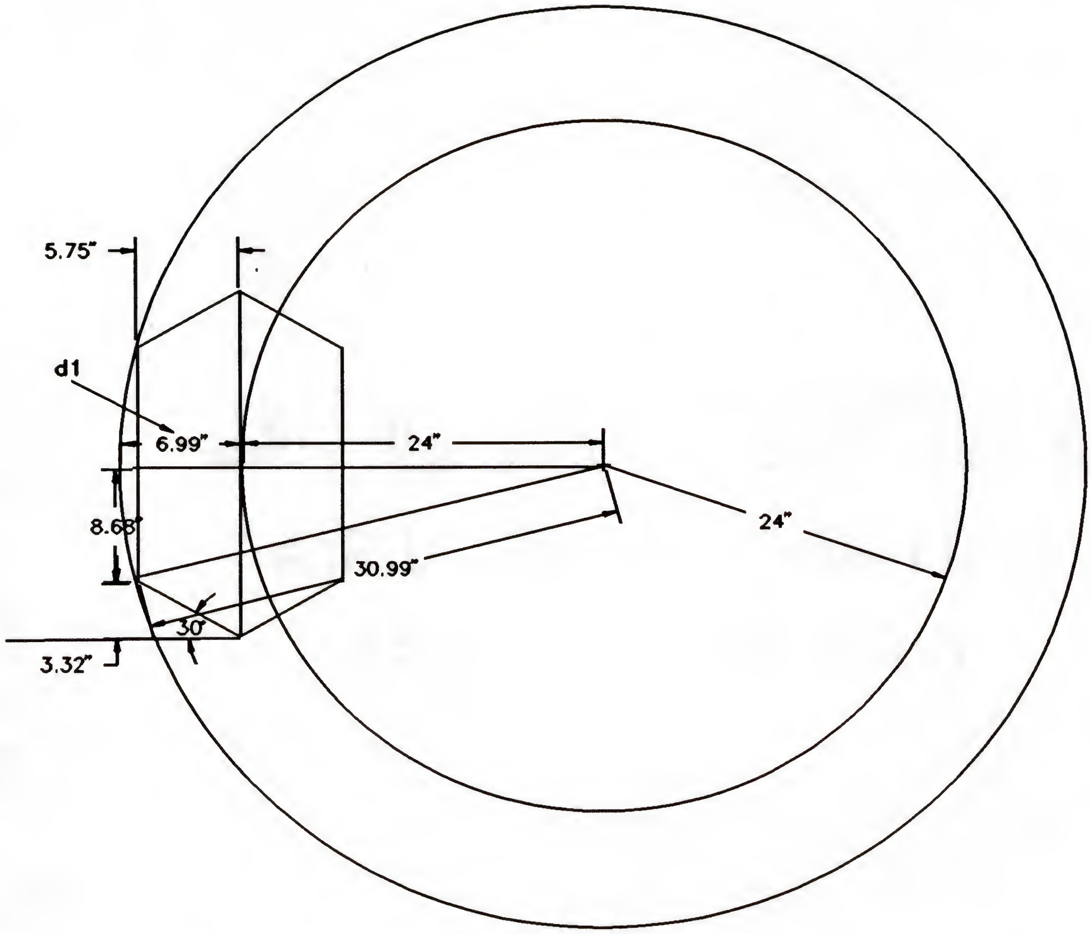


Figure 3.1 Expansion for Odd Segment

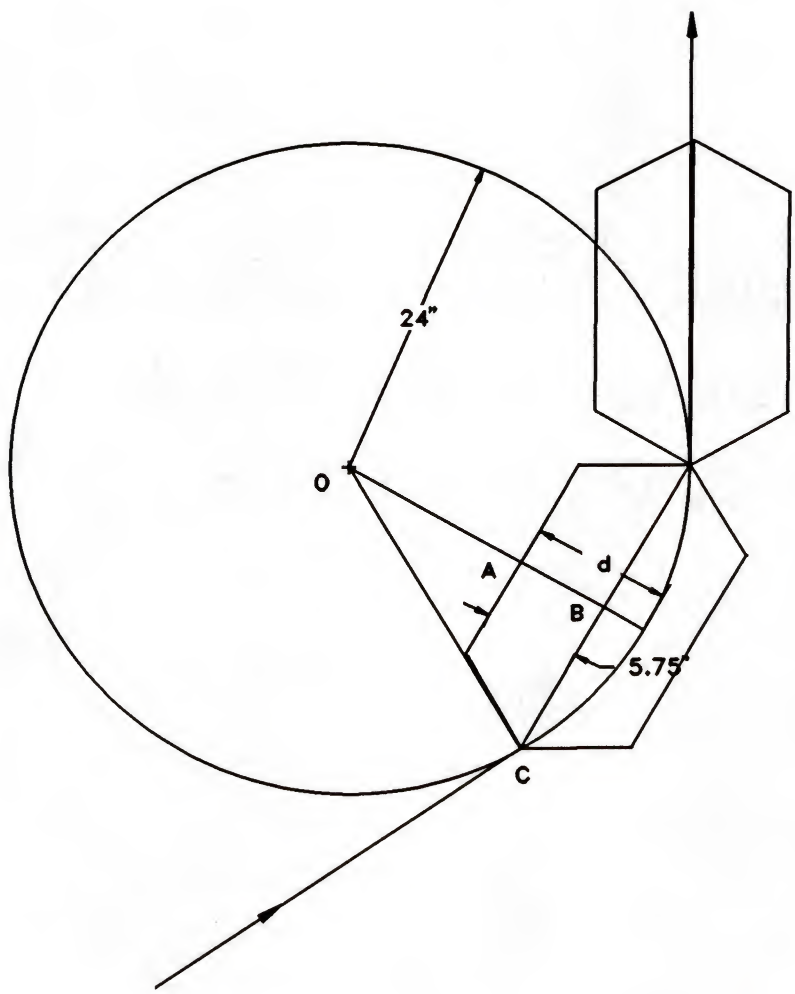


Figure 3.2 Expansion for Even Segment

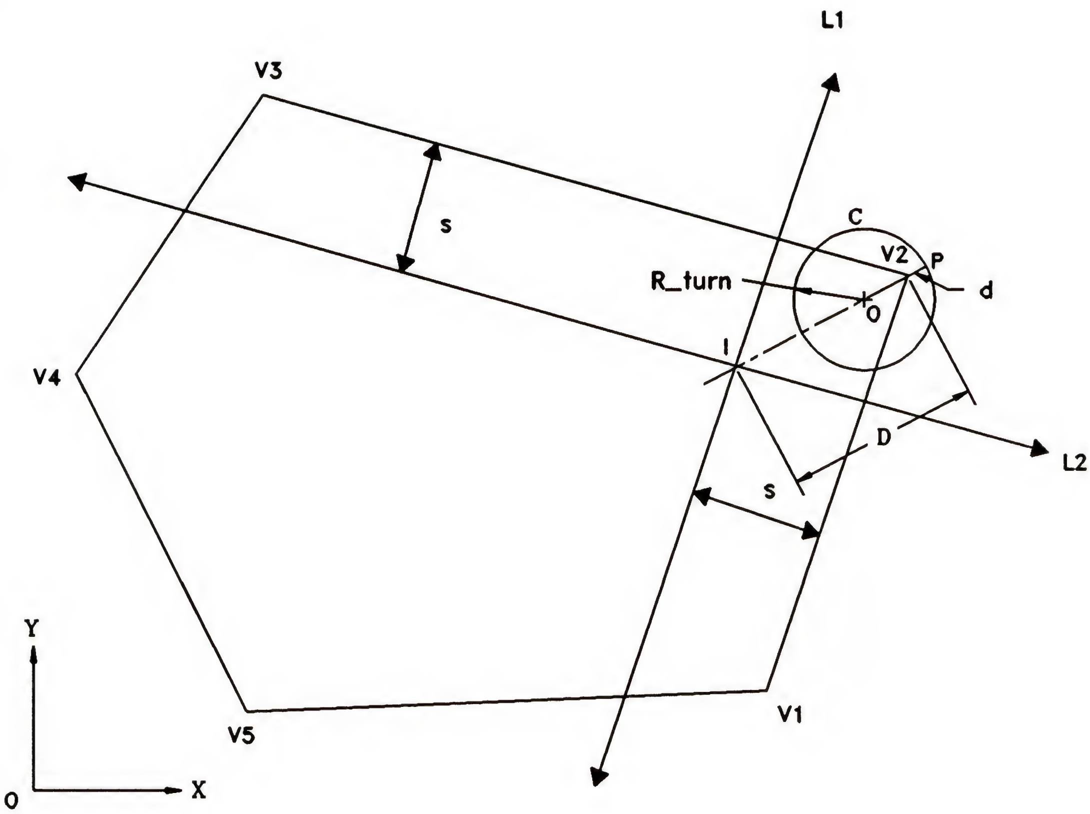


Figure 3.3 Expansion Circle

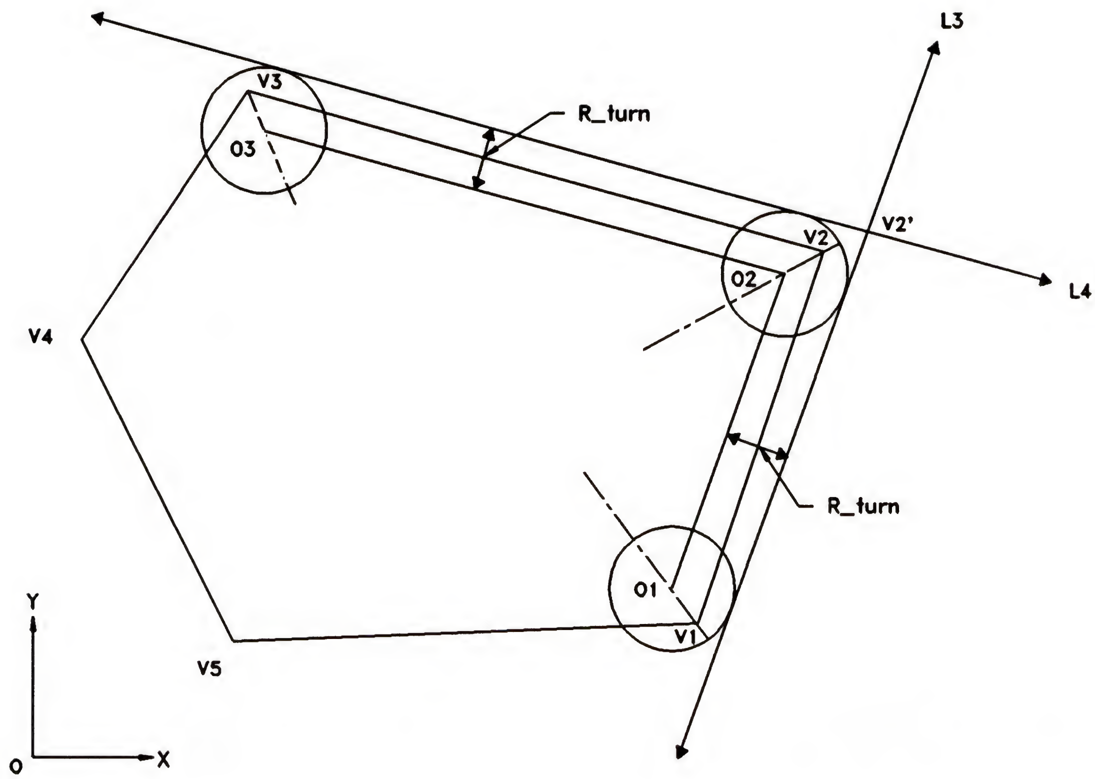


Figure 3.4 Expansion of the edge of the Obstacle

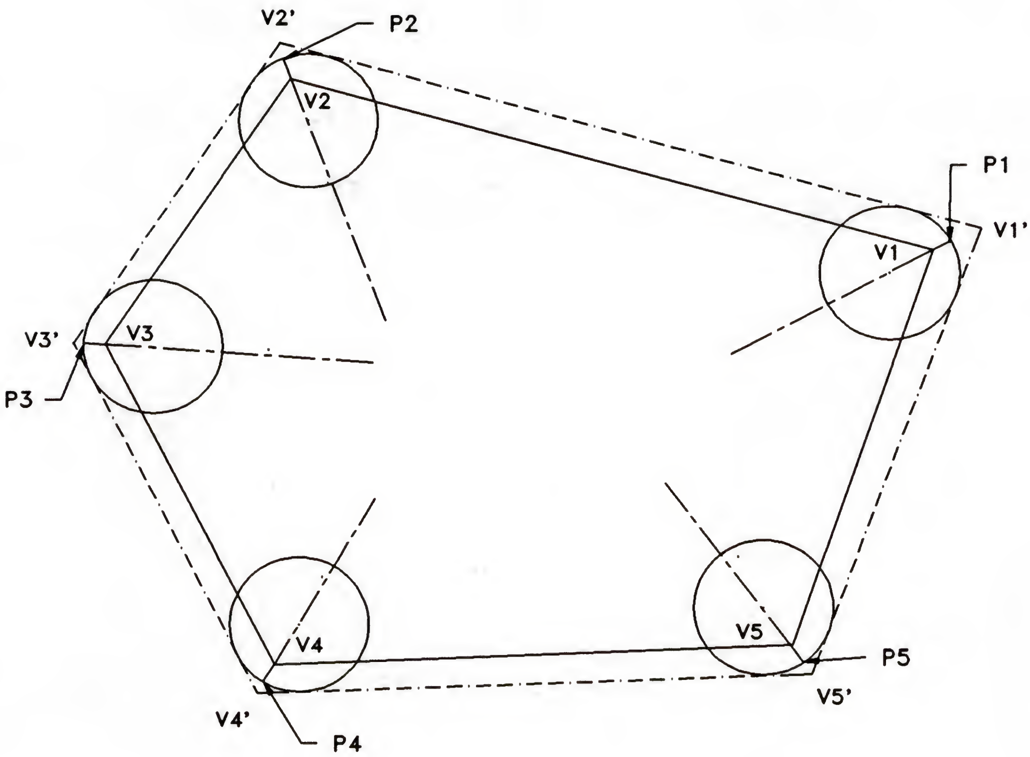


Figure 3.5 Expanded Obstacle

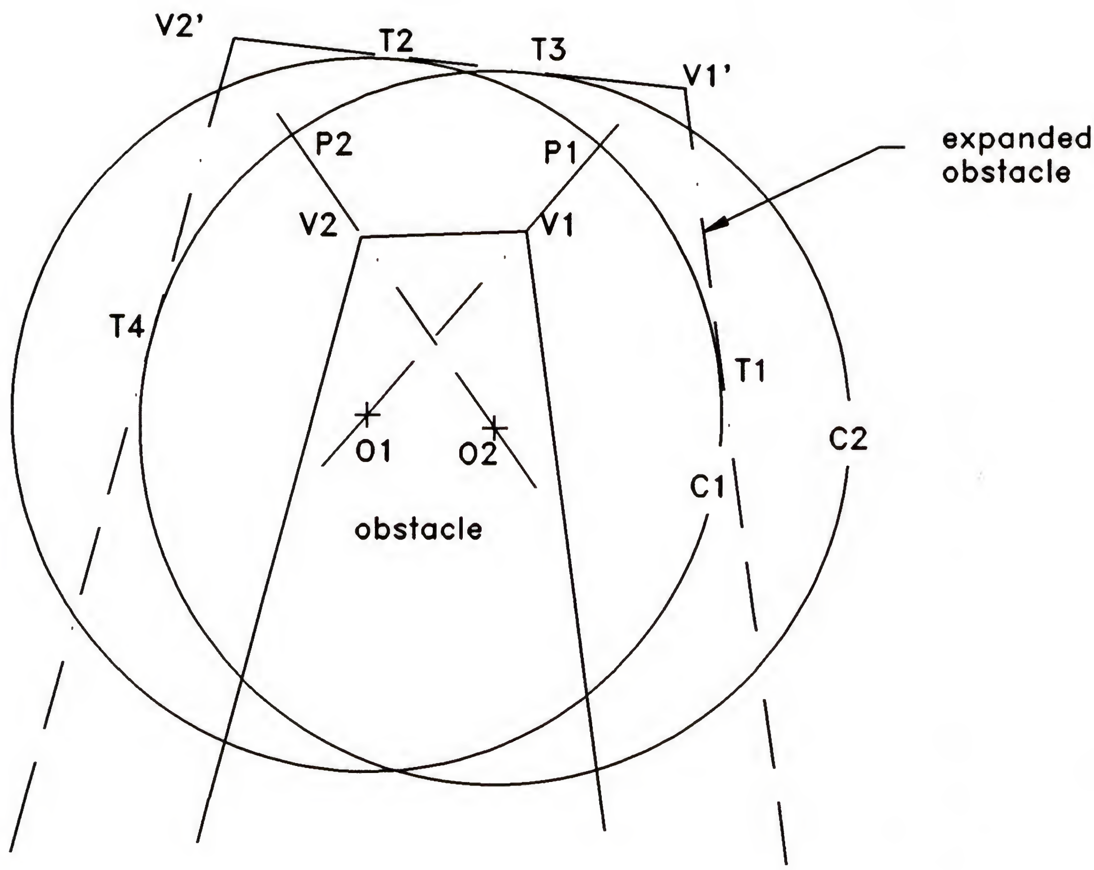


Figure 3.6 Obstacle with Short Edge

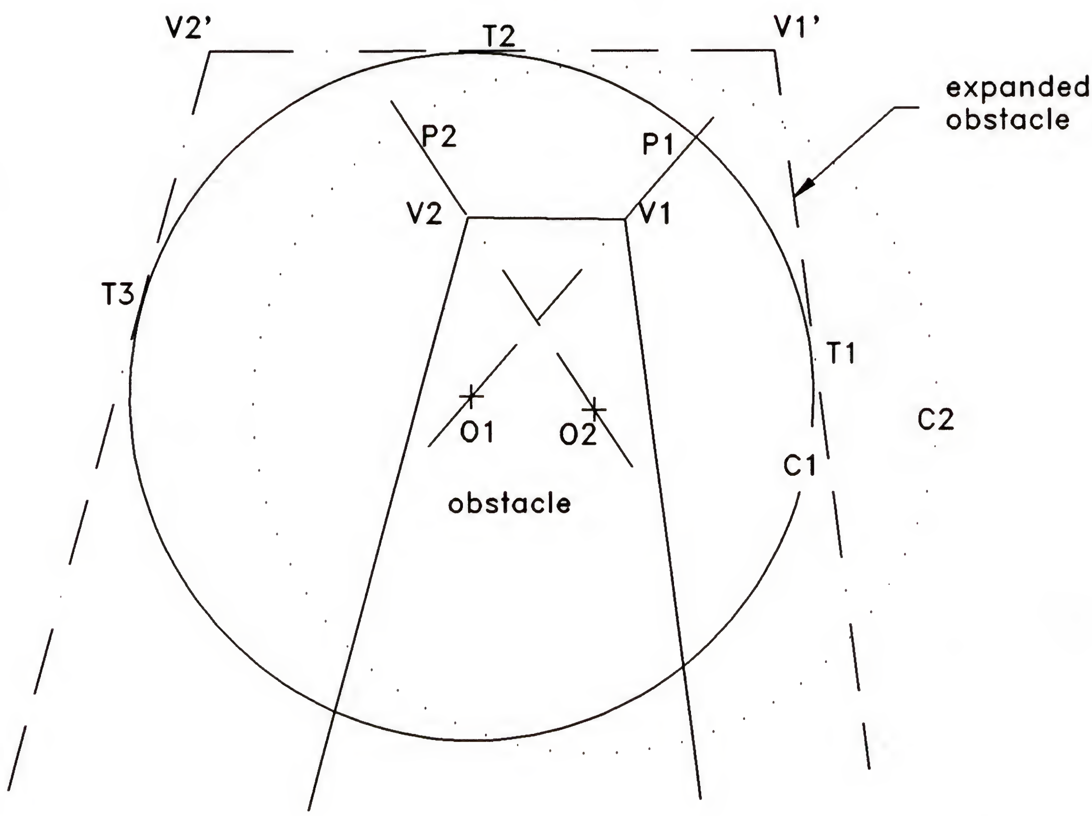


Figure 3.7 Expansion Circle for Obstacle with Short Edge

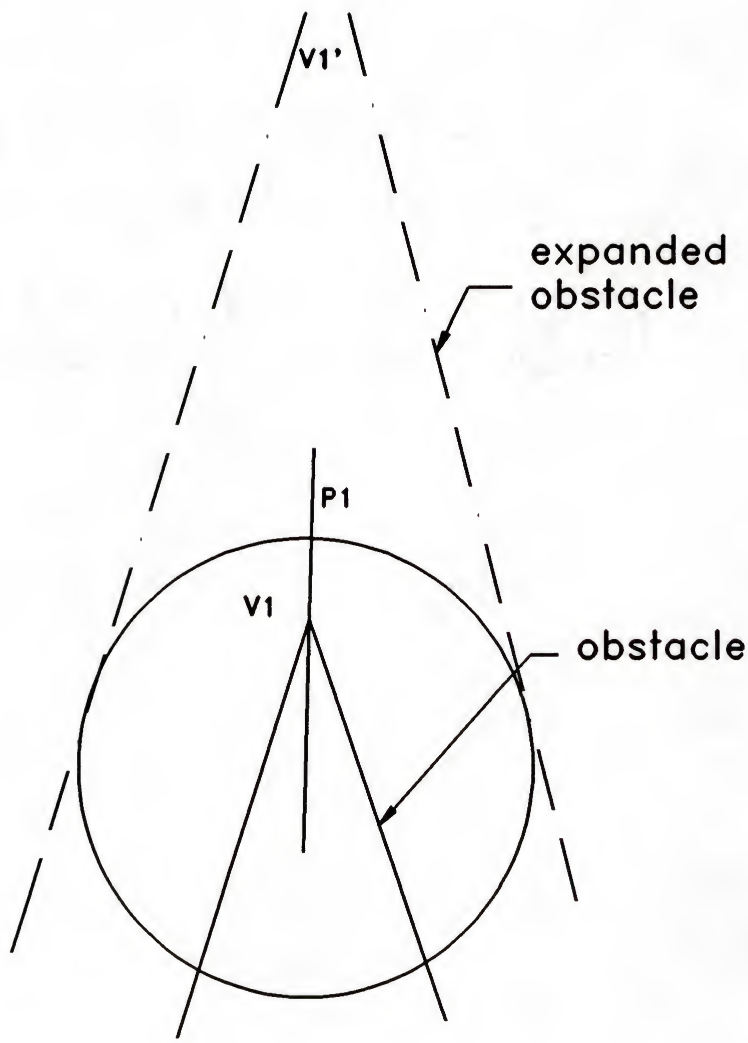


Figure 3.8 Obstacle with Sharp Corner

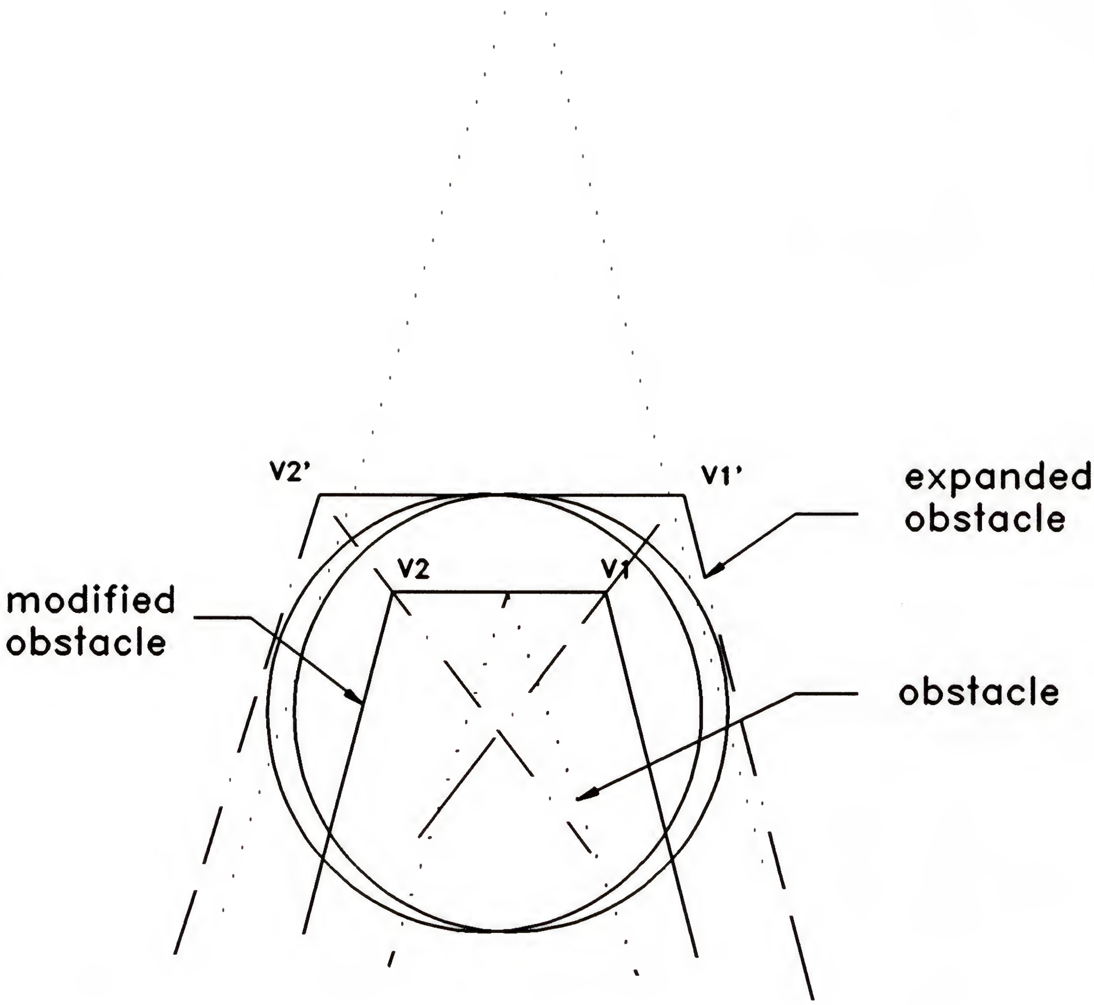


Figure 3.9 Modification of Sharp Corner of the Obstacle

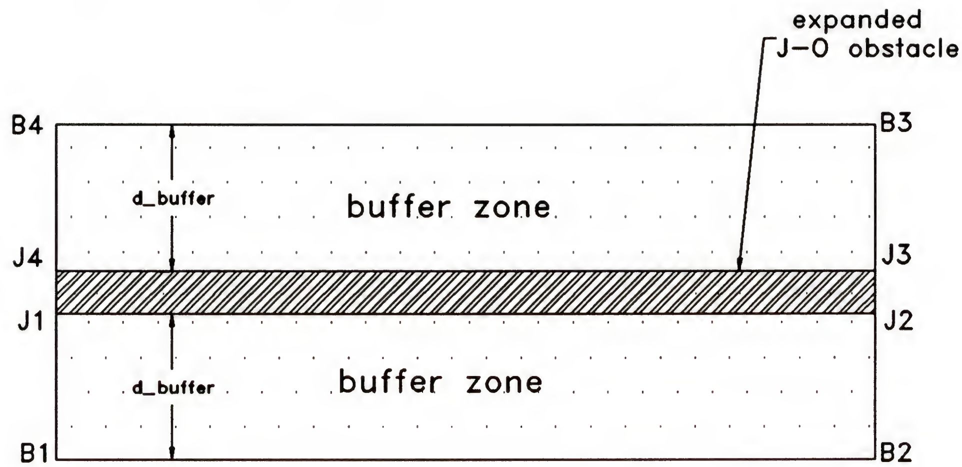


Figure 3.10 Buffer Zones for the J-O Obstacle

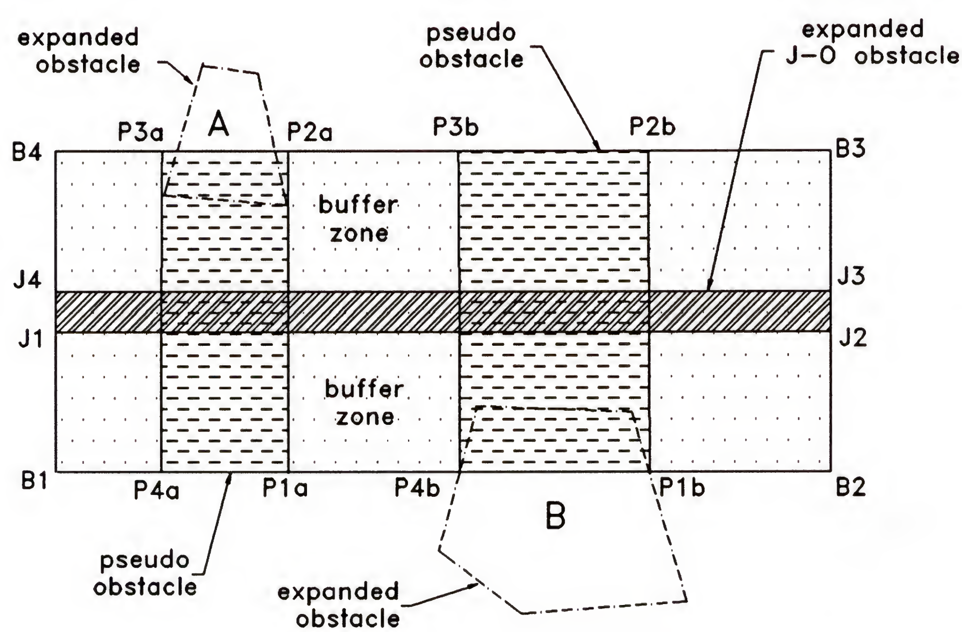


Figure 3.11 Representation of J-O Obstacle (top view)

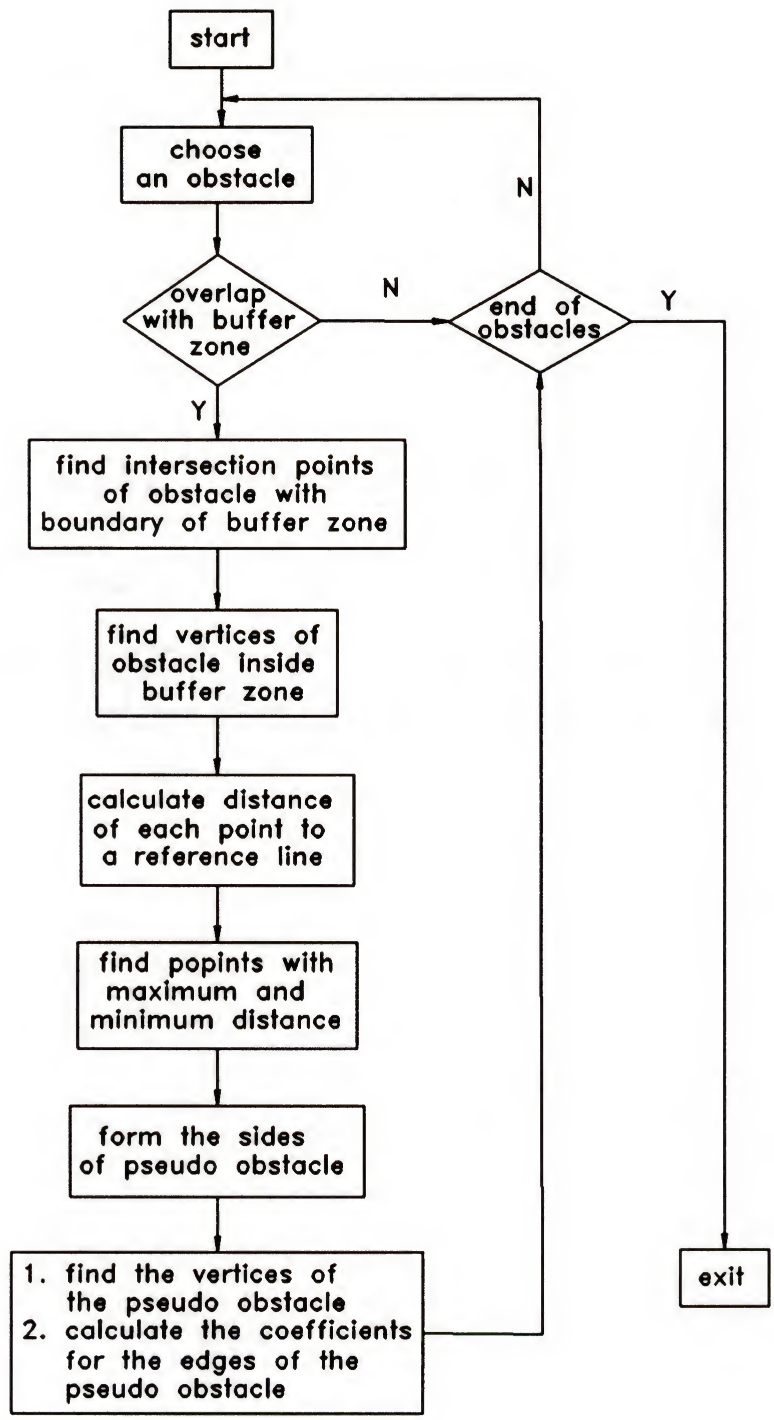


Figure 3.12 Diagram of Creating Pseudo Obstacles

CHAPTER 4
HIERARCHICAL PATH PLANNING

The designed horizontal and vertical motion capabilities of the ATMS enable it to travel on a horizontal plane, and between horizontal planes which are in different elevations. The path planning for the ATMS can be broken down into two major tasks. First, decide the horizontal planes that the ATMS will travel through from the starting plane to the goal plane. Second, plan a collision-free path between two designated positions on a horizontal plane. A hierarchical structure of path planning is believed to be the best choice of approach for this case. Figure 4.1 shows the diagram of the hierarchical structure of path planning.

In the first level of path planning, **vertical planning**, a sequence of via planes from the given starting plane to the goal plane is planned based on the knowledge of the reachable relationship among planes in the environment. In the second level of path planning, **horizontal planning**, a nominal horizontal path from the sub-starting point to the sub-goal point is planned for each of the via planes. The nominal horizontal path is essentially a series of connected straight line segments on the via planes. The path planning at this level is based on the knowledge of the obstacles on the plane.

In the third level of path planning, **operative planning**, the nominal horizontal path on each of the via plane is modified to meet the requirement of minimum turning radius for the horizontal motion and planning a vertical path over J-O obstacles. Another task of operative planning is to plan a vertical path between two horizontal planes to connect the modified paths on each of the horizontal planes. Finally, path data of the complete collision-free path are stored in the database of the system for future use in computing the motion data.

4.1 Vertical Planning

The objective of vertical planning is to find a sequence of horizontal planes that the ATMS will pass through between the given starting and goal planes. An adjacency matrix of planes is created according to the reachable relationship of the horizontal planes in the working environment. Based on the adjacency matrix and the cost from one plane to the other, a cost matrix of planes can be formed. A via matrix of planes can be established base on the cost matrix of plane. From the via matrix of planes, the optimum via planes can be easily determined.

The input and output for vertical planning, and knowledge needed from the database of the system for this level of planning are stated as follows:

Input

The planes where the starting and goal points are located.

Knowledge needed for planning

1. Reachable relationship among planes.
2. For each plane:

   Access ways to other planes, which includes:

   a. Positions of the take-off point to every reachable plane.
   b. Positions of the landing point from other planes.

Output

A sequence of via planes from the starting plane to the goal plane.

4.1.1 Adjacency Matrix for Vertical Planning

The working environment of the ATMS consists of several planes which are in different elevations (see Figure 4.2). The access ways between planes are known a priori. This plane information can be represented by a graph (Figure 4.3) which consists of a set of nodes and a set of arcs. Each arc in the graph is specified by a pair of nodes. The node represents the plane and the arc represents the access way between two planes. A common implementation of a graph is the adjacency matrix. An adjacent matrix for a graph with nodes numbered 1,2,...,N is an N by N Boolean array such that A[i,j] equals

TRUE if and only if there is an arc from node i to node j. Figure 4.4 shows the adjacency matrix for the graph in Figure 4.3. In the adjacency matrix, the element A[i,j] has a value of 1 to indicate that there is an access way between plane i and plane j; otherwise, A[i,j] has a value of 0.

4.1.2 Cost Matrix for Vertical Planning

Based on the adjacency matrix and the cost from one horizontal plane to the other, a cost matrix of the planes can be formed. The cost can be any factor that needs to be minimized (e.g., maximum torque needed for traveling from one plane to another, the heuristic difficulty of travel on the horizontal plane, etc.) for the ATMS to travel from one plane to another. The maximum torque needed for the ATMS to travel from one plane to another plane is used as the example to form the cost matrix.

The torque needed for the ATMS to move from one horizontal plane to another depends on the designed pattern of the vertical path. The vertical path shown in Figure 4.5 is designed under the condition that the side (or wall) of the L-O obstacle is vertical. For the case that the side of the L-O obstacle is not vertical or the access way is not vertical, a different pattern of vertical path is designed as shown in Figures 4.6 and 4.7. In these cases, the maximum torque occurs at point P during the vertical motion, and it can be

calculated as

$$Torque=\left[\frac{L_{seg}^{2}}{2}+(2r\theta+L)\left(r\sin\theta+\frac{L\cos\theta}{2}+L_{seg}\right)+\right.$$
$$\left.(L_a+L_c)\left(\frac{L_a+L_c}{2}+2r\sin\theta+L\cos\theta+L_{seg}\right)\right]*q \tag{4.1}$$

where L_{seg}, L_a, L_b, and L_c have the dimensions shown in Figure 1.1, r is the required turning radius of the ATMS, and q is the distributed weight of the segment (lb/in).

Figure 4.8 shows the curves of the maximum torque needed for the ATMS to move between horizontal planes with different heights (h) and inclination angles (θ). This figure is drawn with a value for q equal to 4.17 lb/inch.

The cost matrix can be established by using Eq.(4.1) to calculate the torque needed for the ATMS to move from one horizontal plane to another. In the process of establishing the cost matrix, the joint capacity of the ATMS is also considered. If a calculated torque exceeds the joint capacity of the ATMS, an infinite number is assigned to the corresponding place in the cost matrix to indicate that the vertical motion between those two horizontal planes is not possible. The maximum torque needed for the ATMS to travel from plane i to plane j is calculated as the cost c_{ij} in the cost matrix **C** in Figure 4.4. A maximum allowable torque of 40,000 in-lb was used in the figure.

4.1.3 Via Matrix for Vertical Planning

The cost matrix contains the information about the cost of moving from one horizontal plane to another. This information is expressed in a matrix form which can be manipulated easily to create a via matrix. The via matrix provides the information about choosing the via plane that will cost least for the ATMS to move from one horizontal plane to another. The procedure to create a via matrix based on the cost matrix for a n-plane environment is described as follows:

1. Initialize and assign the element v_{ij} of the via matrix **V** equal to 0.
   LOOP1:
2. Set the starting plane i from plane 1 to plane n.
   LOOP2:
3. Set the ending plane j from plane 1 to plane n.
   LOOP3:
4. Set the plane k from plane 1 to plane n.
5. Calculate the total cost c_{ikj}, which is the cost for the ATMS to move from plane i through plane k to plane j.
   $$c_{ikj} = c_{ik} + c_{kj}.$$
   where c_{ik} and c_{kj} are the elements in cost matrix.
6. Compare c_{ikj} with the element c_{ij} of the cost matrix.
   if $c_{ikj} < c_{ij}$
   {
   replace c_{ij} with c_{ikj};
   assign the plane number k to v_{ij};
   }
   else;
7. Go to **LOOP3**.
8. Go to **LOOP2**.
9. Go to **LOOP1**.

The element v_{ij} of the via matrix has a value of either the number of a plane or 0. An element v_{ij} which has a value

other than 0 means that plane v_{ij} is the via plane for the ATMS to move from plane i to plane j with the least cost. An element v_{ij} having a value equal to 0 indicates that there is no via plane between plane i and plane j, and the ATMS moves directly from plane i to plane j with the least cost.

For the given starting and goal planes, a sequence of via planes which has least cost can be determined according to the via matrix. For example, the via matrix shown in Figure 4.4 is created based on the cost matrix for the environment in Figure 4.2. To find the via planes from plane 2 to plane 5, first, the value of element v_{25} of the via matrix is examined. v_{25} has a value of 4 meaning that plane 4 is the via plane from plane 2 to plane 5. Second, the values of elements v_{24} and v_{45} are examined and they all have the value of 0, which means that there is no via plane from plane 2 to plane 4 and from plane 4 to plane 5. Plane 4 is the via plane found for the ATMS to move from plane 2 to plane 5. The final result of the vertical planning for this example is from plane 2 through plane 4 to plane 5.

4.2 Horizontal Planning

A sequence of via planes is planned in the vertical planning. The path planning is now reduced to plan a collision free path on the via planes. The input for this level of planning, the knowledge needed from the database of

the system, and the output from this level of planning are stated as follows:

Input

The sequence of via planes which has been planned in the vertical planning.

Knowledge needed for planning

1. The sub-starting point and sub-goal point on each of the via planes. The sub-starting point will be the landing point from the preceding plane to the current plane and the sub-goal point will be the take off point from the current plane to the next plane.
2. The orientation of the ATMS at the take off and landing points.
3. The descriptive and quantitative information about obstacles on each of the via planes.

Output

The nominal horizontal path on each of the planes.

For a G-A obstacle on the horizontal plane, only horizontal motion is possible for the ATMS to pass it. However, for a J-O obstacle, both horizontal and vertical motion are considered for the ATMS to pass it. The path segments, which are planned in this level, are line segments on the horizontal plane. Even the vertical path segment for jumping over a J-O obstacle is just a line segment on the

horizontal plane. The planned vertical path segment for jumping over a J-O obstacle is called a proposed vertical path. It is shown in this section that both the horizontal path and the proposed vertical path are planned by using the same planning algorithm.

4.2.1 Planning for a Horizontal Path

Planning a path for horizontal motion of the ATMS is the major concern in this section. All the obstacles, including the J-O obstacles, are treated as G-A obstacles. The algorithm of horizontal planning which is developed in this section will also be applied to plan a proposed vertical path later in section 4.2.2.

In the obstacle expanded environment, the ATMS is considered as a point and the path planning can be achieved by a graph-search approach. A general graph-search procedure is described as follows:

1. Create a search graph, G, consisting only of the starting node s. Put it on a list called OPEN.
2. Create a list called CLOSE that is initially empty.
3. LOOP: if OPEN is empty, exit with failure.
4. Select the first node n on OPEN, remove it from OPEN and put it on CLOSED.
5. If the node n is the goal node, exit successfully with the path by tracing back to the starting node s.

6. Expand node n based on the defined expanding rules. Put these successors of node n in a set M.
7. Calculate the cost of the successors of node n according to the defined evaluation function.
8. Establish 'father-son' relationship for node n with those members of M which are not on lists OPEN and CLOSED. For those members of M which are already on either of OPEN or CLOSED, decide whether or not to redirect its parent and update the cost.
9. According to the cost of the node, reorder the node on OPEN in a increasing order.
10. Go to LOOP.

In the algorithm of horizontal planning, the concept of the visible tangent point is used as the rule for expanding the node in step 6. Figure 4.9 shows how to define the visible tangent points. The successors to the point P are those visible tangent points from point P. To reduce the search effort, a heuristic search method, A^* search, is used for horizontal planning. In the A^* search method, a heuristic function f^* is used to estimate the cost of each node and decide the node for next expansion. The heuristic function f^* comprises two parts and can be expressed as

$$f^* = g^* + h^* \tag{4.2}$$

where

g^*: Cost of traveling from starting node to current node. This is an actual cost.

h^*: Estimated cost of traveling from current node to a goal node. This is the carrier of heuristic information.

Based on the A^* search technique and expanding a node by the visible tangent points, the algorithm of horizontal planning for a horizontal path is developed as shown in Figure 4.10. The algorithm is described as follows:

1. Create lists: O_vtx, and C_vtx. Initially, list O_vtx contains only starting point and C_vtx is empty.
2. Connect the starting and goal points by a straight line segment.
3. If goal point is visible, put it on O_vtx.
4. If not visible, find the visible tangent points, estimate their costs according to the heuristic function, and put them on O_vtx in a increasing cost order.
5. Establish pointers to their parent node for these visible tangent points.
6. **LOOP:** if O_vtx is empty, exit with failure.
7. Select the first point on O_vtx, remove it from O_vtx, and put it on C_vtx.
8. If the point is goal point, exit successfully and the path can be obtained by tracing back from the goal point to the starting point. Otherwise, assign the point as new

starting point. Check the visibility from new starting point to goal point.

9. If visible, calculate the cost and put the goal point on O_vtx at the proper position.
10. If not visible, find the visible tangent points and calculate their costs. For those visible tangent points which are already on either O_vtx list or C_vtx list, whether to redirect their pointers to parent or not based on the estimated cost. For those visible tangent points which are not on either O_vtx or C_tx lists, establish the pointers to their parent and insert them on list O_vtx in the proper positions according to their estimated cost.
11. Go to **LOOP**.

The procedures of finding the visible tangent points for a specified point in steps 4 and 10 are shown in Figure 4.11. The tangent points from a point to the obstacle are found first, and then, the visibility of those tangent points is determined. The line equations for the edges of the expanded obstacle are used to find the tangent points for a point to an expanded obstacle, and it is expressed as

$$Ax + By + C = 0. \tag{4.3}$$

The coefficients (A, B , and C) of the line equation for the edge of the expanded obstacle has been calculated and stored in the database of the system.

The vertices of the expanded obstacle A shown in Figure 4.12 are in counter clockwise sense. The coordinates of point P, and lines L_1 and L_2 are used to decide whether or not the current checked vertex V_2' of the expanded obstacle is a tangent point from point P to the expanded obstacle A. The equations of L_1 and L_2 are expressed as

$$L_1: A_1x + B_1y + C_1 = 0,$$
$$L_2: A_2x + B_2y + C_2 = 0. \qquad (4.4)$$

Substituting the coordinates of point P into the line equations expresses them as follows:

$$D_1 = A_1x_P + B_1y_P + C_1,$$
$$D_2 = A_2x_P + B_2y_P + C_2. \qquad (4.5)$$

D_1 and D_1 may have positive, negative, or zero value. The following are the principles used to find tangent points while vertex V_2' is the current checked vertex:

1. If the product of D_1 and D_2 is negative, vertex V_2' is a tangent point.
2. If the product of D_1 and D_2 is positive, vertex V_2' is not a tangent point.
3. If both D_1 and D_2 are equal to 0, vertices V_1' and V_3' are the tangent points.

4. If D_1 is equal to 0 and D_2 is not, the tangent point can be decided based on the sign of D_2,
   a) If D_2 is greater than 0, vertex V_1' is a tangent point.
   b) If D_2 is less than 0, vertex V_2' is a tangent point.
5. If D_2 is equal to 0 and D_1 is not, the tangent point can be decided based on the sign of D_1,
   a) If D_1 is greater than 0, vertex V_3' is a tangent point.
   b) If D_1 is less than 0, vertex V_2' is a tangent point.

There are two and only two tangent points which can be found from a point to a convex polygonal obstacle. The principles stated above are applied to each of the vertices of the obstacle in turn until two tangent points are found. Each of the tangent points is then checked for visibility to decide whether it is the successor to the current point. Checking the visibility of a tangent point from a given point P is actually checking the intersection of the connecting line segment with the obstacles. In order to reduce the computation in checking for intersection, two steps of checking are applied: First, bound the expanded obstacles and the line segment with rectangles as shown in Figure 4.13. By checking the maximum and minimum coordinates of the rectangles, some of the obstacles can be determined as non-intersected and excluded from further checking for

intersection. No computation is needed for the intersection check in this step, only Boolean expressions are used. This saves computation time in the whole process of checking for intersection. Second, check the intersection of the line segment with the prospective expanded obstacle. Basically, in this step of checking, the possible intersection of two line segments is checked. The data of the coefficients of the line equations for the edges of the expanded obstacles has been stored in the database of the system. In case no intersection occurs, the tangent point is "visible" to the point P. The visible tangent point is a successor to point P. After all of the successors to point P have been found, the cost of each of the successors is calculated by the heuristic cost function. The next expansion point is chosen based on the calculated costs of all the leaf nodes in the search tree.

Figure 4.14 shows an example of planning a horizontal path from point S to point G by using visible tangent points and heuristic cost function in a given environment. The three numbers, which are listed beside the vertex, are the actual, heuristic, and total costs evaluated by the cost function. The cost calculated in this example is the distance between two points. The vertex which has the minimum total cost is chosen for next expansion. After three steps of expansion, the goal point G is reached. The horizontal path from point S to point G is planned by tracing back from point G to point S.

4.2.2 Planning for Proposed Vertical Path

In the horizontal planning, a proposed vertical path over a J-O obstacle is also planned. It is necessary to emphasize that the proposed vertical path which is planned in the horizontal planning is a line segment on the horizontal plane and crosses a J-O obstacle. The information of this line segment is used to plan a third dimensional path later in the operative planning. In this section, the means whereby the algorithm for planning horizontal paths can be used to plan a proposed vertical path is introduced.

A vertical path is to be planned when a J-O obstacle is encountered. The representation of a J-O obstacle by the buffer zone and pseudo-obstacle has been presented in section 3.3. The purpose of the representation is to plan a proposed vertical path over the J-O obstacle in the process of horizontal planning.

Two criteria are designed to fit the buffer zone and pseudo obstacle representation of J-O obstacles into the process of horizontal planning:

(a) When a J-O obstacle is intersected, the pseudo obstacles related to the J-O obstacle are included as G-A obstacle in the horizontal planning.

(b) When only a J-O obstacle is intersected, it indicates that a vertical path can be planned in this situation.

As mentioned before, the proposed vertical path planned in

horizontal planning is just a straight line segment on the horizontal plane. It contains the information about the projection of the vertical path on the horizontal plane. This information will help to create a vertical path in the operative planning.

Criterion (a) and (b) are represented by the diagrams shown in Figure 4.15 and Figure 4.16. These two diagrams are inserted at position I and II in the Figure 4.17 for planning a proposed vertical path. After adding these two criteria in the process of finding the visible tangent points, the algorithm of horizontal planning which is based on the A^* graph search technique, can plan both horizontal and proposed vertical paths in the same process.

An example of planning a proposed vertical path over J-O obstacle is shown in Figure 4.18. Line segment $\overline{SG}$ intersects with a J-O obstacle, so the related pseudo-obstacle is included for intersection check, and the pseudo-obstacle O_pseu is found to be intersected. Two tangent points A and C are calculated for the pseudo obstacle O_pseu. Checking the visibility from point S to point C, only the J-O obstacle is intersected, therefore, a proposed vertical path is planned. The proposed vertical path which is planned from T to L is based on the intersection point of $\overline{SC}$ with the related buffer zone. Point T and point L are stored as take-off point and landing point in the database of the system. A link from point S through point T to point L is established. From point

L, the horizontal planning continues. In calculating the cost of point L, a factor is applied on the evaluation function which is used for planning horizontal path. This factor can be decide by the ratio of energy consumption for vertical motion and horizontal motion, or by the tendency of planning a vertical motion or horizontal motion.

4.2.3 Planning for the Start and Goal Orientations

The orientations of the ATMS at the starting and goal positions are not included in the horizontal planning described above. Since a radius is required for the ATMS to turn, the planned path may not be collision-free at the starting or goal point. Figure 4.19 shows the planning of a collision-free path while the orientation is considered. Line segment $\overline{SV_1}$ is the planned path which does not consider the orientation. If the ATMS moves from point S toward point V_1, a circular path $\overset{\frown}{ST}$ is followed first, the path from point T to V_1 intersects with obstacle B. To avoid the collision, the path from point T to point V_1 should be replanned. It is accomplished simply by performing another horizontal planning from point T to point V_1. The final path, in which the orientation is also considered, is a path from point S through points T and N to point V_1.

The path which is planned in horizontal planning is a series of line segments and it is not suitable for the ATMS to

follow. The path is called the "nominal horizontal path". A linked list of the points on the nominal horizontal path is established from the starting point to the goal point. The coordinates of the point and point type are stored in each member of the list. The point type is used to distinguish a take off point and a landing point from a normal point of the horizontal path. On the list of a take off point, the width and height of the related J-O obstacle are also stored.

4.3 Operative Planning

The objective of this level of planning is to modify the nominal horizontal path on the horizontal plane and plan vertical paths between planes to connect the modified horizontal paths to form an overall collision-free path. The modification of the nominal horizontal path includes:

1. Planning circular arcs at the turning points of the path.
2. Planning vertical path segments over J-O obstacles.

After the modifications, the path on each of the horizontal planes is suitable for the ATMS to follow. Vertical paths between horizontal planes are planned to complete the overall collision free path.

4.3.1 Plan a Circular Path at Turning Point

The nominal horizontal path is a series of connected line segments. A circular arc is planned to smoothly join the two connected line segments and satisfy the turning radius requirement of the ATMS. From the starting point of the nominal horizontal path, every three consecutive points are used to find the circular path between the two connected line segments.

In Figure 4.20, points $N_1(x_{N1}, y_{N1})$, $N_2(x_{N2}, y_{N2})$, and $N_3(x_{N3}, y_{N3})$ are the three consecutive points on the nominal horizontal path. It is assumed that point N_1 is not a take off point. Knowing the coordinates of the three points, the direction cosine of the lines N_1N_2 and N_2N_3 can be calculated and expressed as (o_{1x}, o_{1y}) and (o_{2x}, o_{2y}). Lines L_1 and L_2 can be found by parallel shifting lines N_1N_2 and N_2N_3 by the distance R_turn. The equations of L_1 and L_2 are expressed as

$$L_1: \quad A_1x + B_1y + C_1 = 0,$$

$$L_2: \quad A_2x + B_2y + C_2 = 0. \qquad (4.6)$$

The coordinates of the intersection point O of lines L_1 and L_2 can be calculated by using Cramer's rule,

$$x_0 = \frac{\begin{vmatrix} B_1 & C_1 \\ B_2 & C_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}}$$

$$y_0 = \frac{\begin{vmatrix} C_1 & A_1 \\ C_2 & A_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} \tag{4.7}$$

A circle C which is centered at point O, has a radius of R_turn and tangent to both line segments $\overline{N_1N_2}$ and $\overline{N_2N_3}$ can be drawn. The arc $\overset{\frown}{T_1T_2}$ between these two tangent points is the modified path for the ATMS to turn at point N_2. The distance d_1 can be calculated by the distances s_1 and R_turn. The coordinates of the tangent point on the line segment $\overline{N_1N_2}$ are calculated by

$$x_{T1} = x_{N1} + O_{1x} * d_1,$$
$$y_{T1} = y_{N1} + O_{1y} * d_1. \tag{4.8}$$

Similarly, the coordinates of the tangent point T_2 can be calculated. After the coordinates of tangent points T_1 and T_2 have been calculated, the angle (θ) of the circular arc $\overset{\frown}{T_1T_2}$ can be calculated by

$$\theta = \cos^{-1}(O_{1x} * O_{2x} + O_{1y} * O_{2y}) \qquad (0^\circ <= \theta <= 180^\circ) \qquad (4.9)$$

After the turning angle has been calculated, the direction of turn must be decided. The value of the determinant D, which is formed by the coordinates of points O, T_1, and T_2, is used to decide the direction of turn. The determinant is expressed as follows,

$$D = \begin{vmatrix} x_o & y_o & 1 \\ x_{T1} & y_{T1} & 1 \\ x_{T2} & y_{T2} & 1 \end{vmatrix} \qquad (4.10)$$

It is defined that if D is positive, the direction of turn is in counter clockwise sense, and the angle θ has a positive value. If D is negative, the direction of turn is in clockwise sense, and the angle θ has a negative value.

The modification plans a circular path between two connected straight line paths. The coordinates of the end points for each of the path segments is calculated. The angle and the direction of turn are also decided for the circular path segment. These data are stored and used to calculate the motion data for the ATMS.

4.3.2 Planning the Vertical Path Over J-O Obstacle

In the modification of the nominal horizontal path, if the starting point of each of the three points is a take off

point, a vertical path is planned over the J-O obstacle. The line TL in the top view drawing of Figure 4.21, is a proposed vertical path across a J-O obstacle. Point T is the take off point and L is the landing point. In the data list of the take off point, the height of the related J-O obstacle is stored. By using this information, the vertical path over the J-O obstacle is planned.

The direction cosine of line TL, (O_x, O_y, O_z), can be calculated by using the coordinates of points T and L. The angle θ is calculated by

$$\theta = 2 * [\sin^{-1}(h/4R\_turn)^{1/2}] \qquad (0^\circ <= \theta <= 90^\circ) \qquad (4.11)$$

The path at T has been modified, point S is the end point of the circular path and also the starting point of the vertical path. The distance (d_1) between T and S is calculated. The distance between point S and point T_1 is calculated by

$$d = d\_buf - L_a - 2 * R\_turn * \sin\theta. \qquad (4.12)$$

By using the distance d and the direction cosine of line TL, the coordinates of point T_1 (x_{T1}, y_{T1}, z_{T1}) are calculated by

$$x_{T1} = x_S + d * O_x,$$
$$y_{T1} = y_S + d * O_y,$$
$$z_{T1} = z_S + d * O_z. \qquad (4.13)$$

After the coordinates of point T_1 are calculated, the coordinates of points T_2 and T_3 can be calculated. The coordinates of point T_2 (x_{T2}, y_{T2}, z_{T2}) is calculated by

$$
\begin{aligned}
x_{T2} &= x_{T1} + R\_turn * \sin\theta * o_x, \\
y_{T2} &= y_{T1} + R\_turn * \sin\theta * o_y, \\
z_{T2} &= z_{T1} + R\_turn * (1 - \cos\theta).
\end{aligned} \tag{4.14}
$$

The coordinates of point T_3 (x_{T3}, y_{T3}, z_{T3}) are calculated by

$$
\begin{aligned}
x_{T3} &= x_{T2} + R\_turn * \sin\theta * o_x, \\
y_{T3} &= y_{T2} + R\_turn * \sin\theta * o_y, \\
z_{T3} &= z_{T2} + R\_turn * (1 - \cos\theta).
\end{aligned} \tag{4.15}
$$

The coordinates of the transition points of the vertical path segments on the landing side of the J-O obstacle can be calculated in the similar way. First, calculate the coordinates of point T_6, then T_5 and T_4. For each of the circular path segments, the coordinates of the center of the circular arc can be easily determined after knowing the coordinates of the transition points. For example, the coordinates of center C_2,(x_{C2}, y_{C2}, z_{C2}) can be found by:

$$
\begin{aligned}
x_{C2} &= x_{T3}, \\
y_{C2} &= y_{T3}, \\
z_{C2} &= z_{T3} - R\_turn.
\end{aligned} \tag{4.16}
$$

By the method described above, a vertical path is planned over the J-O obstacle from its corresponding proposed vertical path. After this modification, the vertical path ends at point T_6, and from this point the operative planning continues.

4.3.3 Planning the Vertical Path Between Planes

A vertical path is planned between horizontal planes to connect the modified path on each of the horizontal planes. The patterns of vertical paths for the ATMS to move up to a higher level or move down to a lower level are shown in Figures 4.22 and 4.23. For the case where the ATMS moves up to a higher level as shown in Figure 4.22, a local coordinate system is introduced. The origin of the local coordinate system is located at point E and the XZ plane is coincided with the vertical path plane, the X-axis is in the horizontal direction shown as in Figure 4.22, the direction of the Z-axis is perpendicular to the horizontal plane. The coordinates of point S (x_s, y_s, z_s) in the local coordinate system can be found through a coordinate transformation. The angle θ and the length L are calculated by

$$\theta = \tan^{-1}\left(\frac{h}{-x_s}\right) \qquad (0^o \leq \theta \leq 90^o) \tag{4.17}$$

$$L = \frac{[h - 2.0*R\_turn*(1-\cos\theta)]}{\sin\theta} \tag{4.18}$$

The coordinates of the transition points in the local coordinate system are expressed as follows:

T_5:

$x_{T5} = L_a \qquad y_{T5} = 0.0 \qquad z_{T5} = d.$

T_4:

$x_{T4} = -L_c \qquad y_{T4} = 0.0 \qquad z_{T4} = d.$

T_3:

$x_{T3} = x_{T4} - R\_turn * \sin\theta$

$y_{T3} = 0.0$

$z_{T3} = z_{T4} - R\_turn * (1 - \cos\theta).$

T_2:

$x_{T2} = x_{T3} - L * \cos\theta$

$y_{T2} = 0.0$

$z_{T2} = z_{T3} - L * \sin\theta.$

T_1:

$x_{T1} = x_{T2} - R\_turn * \sin\theta$

$y_{T1} = 0.0$

$z_{T1} = z_{T2} - R\_turn * (1 - \cos\theta).$

where d is the half height of the ATMS, and La, Lc are dimensions of the ATMS shown in Figure 1.1.

There are two circular path segments in the vertical path, and the coordinates of the centers of the circular path segments expressed in the local coordinate system are as follows:

C_1:

$x_{C1} = x_{T1}$

$y_{C1} = y_{T1}$

$z_{C1} = z_{T1}$ + R_turn.

C_2:

$x_{C2} = x_{T4}$

$y_{C2} = y_{T4}$

$z_{C2} = z_{T4}$ - R_turn.

Similarly, for the case of moving down to a lower level, a local coordinate system is introduced as shown in Figure 4.23. The coordinates of point E (x_E, y_E, z_E) in local coordinate system can be found through a coordinate transformation. The angle θ and the length L are calculated by

$$\theta = \tan^{-1}\left(\frac{h}{x_E}\right) \qquad (0^\circ \le \theta \le 90^\circ) \tag{4.19}$$

$$L = \frac{[h - 2.0*R\_turn*(1-\cos\theta)]}{\sin\theta} \tag{4.20}$$

The coordinates of the transition points in the local coordinate system are expressed as follows:

T_1:

$x_{T1} = 0.0$ $\qquad y_{T1} = 0.0$ $\qquad z_{T1} = d$.

T_2:

$x_{T2} = L_b$ $\qquad y_{T2} = 0.0$ $\qquad z_{T2} = d$.

T_3:

$x_{T3} = x_{T2}$ + R_turn * $\sin\theta$

$y_{T3} = 0.0$

$z_{T3} = z_{T2}$ - R_turn * (1 - $\cos\theta$).

T_4:

$$x_{T4} = x_{T3} + L * \cos\theta$$

$$y_{T4} = 0.0$$

$$z_{T4} = z_{T3} - L * \sin\theta.$$

T_5:

$$x_{T5} = x_{T4} + R\_turn * \sin\theta$$

$$y_{T5} = 0.0$$

$$z_{T5} = z_{T4} - R\_turn * (1 - \cos\theta).$$

Where, d is the half height of the ATMS, Lb is the dimension of the ATMS shown in Figure 1.1.

The coordinates of the centers of the circular path segments expressed in the local coordinate system are as follows:

C_1:

$$x_{C1} = x_{T2}$$

$$y_{C1} = y_{T2}$$

$$z_{C1} = z_{T2} - R\_turn.$$

C_2:

$$x_{C2} = x_{T5}$$

$$y_{C2} = y_{T5}$$

$$z_{C2} = z_{T5} + R\_turn.$$

The dimensions of L_a, L_b, and L_c used in generating the vertical path are for the purpose of ensuring proper take off and landing for the ATMS. The calculated transition points and center of the circular paths are transformed from the local coordinate system to the global coordinate system.

4.4 The Final Planned Collision Free Path

The final path after the modifications is a three-dimensional path which is a series of connected straight line or circular path segments. The data of each of the path segments are stored in the database of the system for future calculation of the motion data. The stored data are

a. Straight line path segment:

1. type of the straight line segment, horizontal path or vertical path,
2. the coordinates of the end points of the path segment,
3. the length of the path segment.

b. Circular path segment:

1. type of the circular path segment, horizontal path or vertical path,
2. the coordinates of the end points of the path segment,
3. the coordinates of the center of the circular path segment,
4. the angle of the circular path segment (direction of turn is expressed by the sign of the angle).

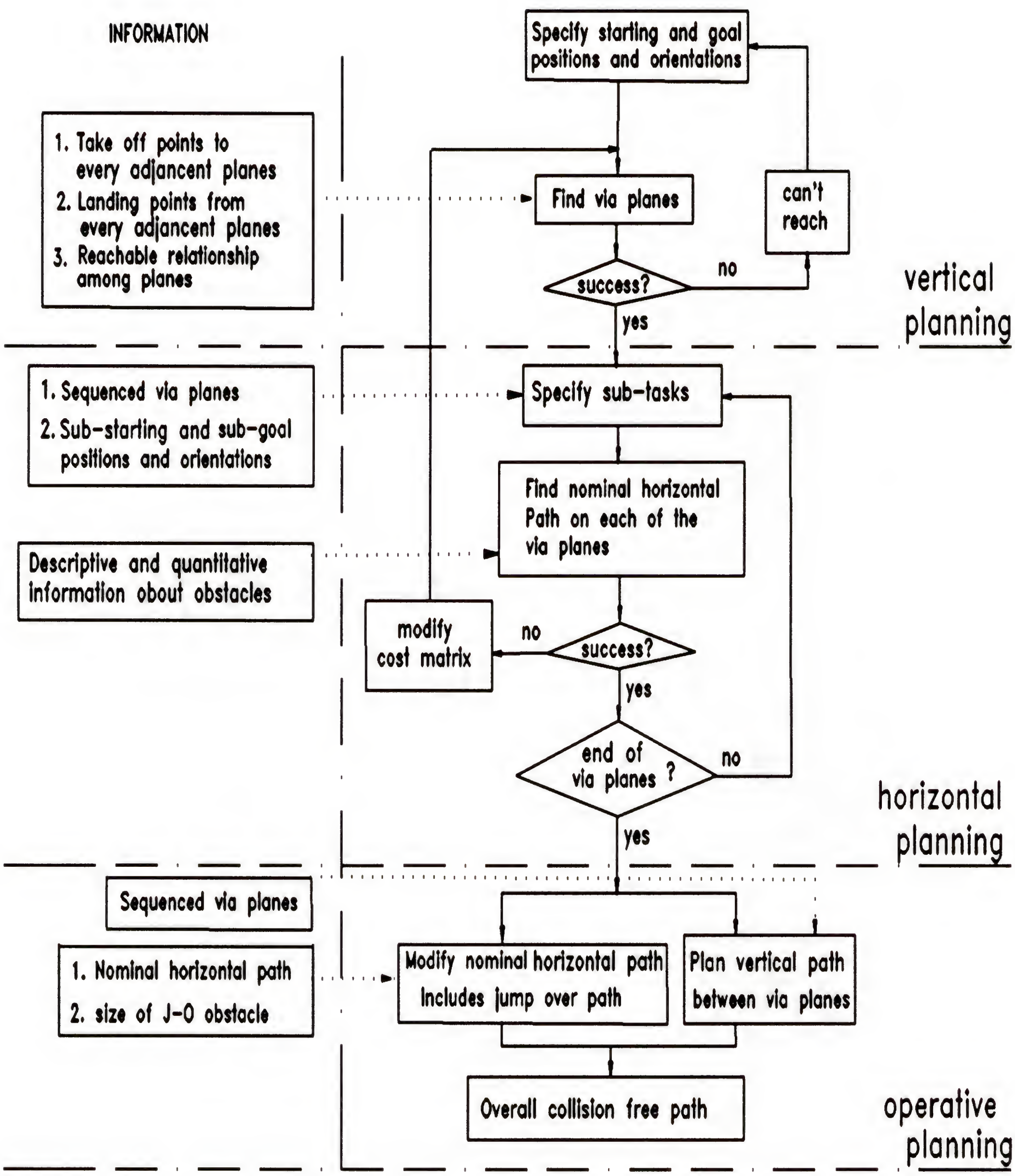


Figure 4.1 Hierarchical Structure of Path Planning

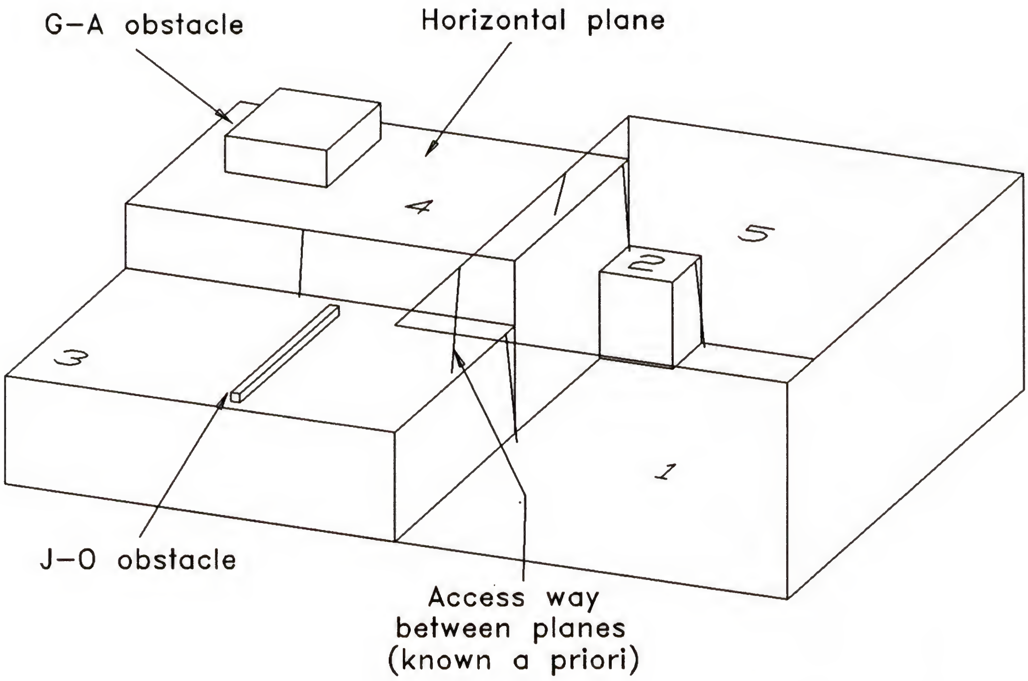


Figure 4.2 Example of ATMS Environment

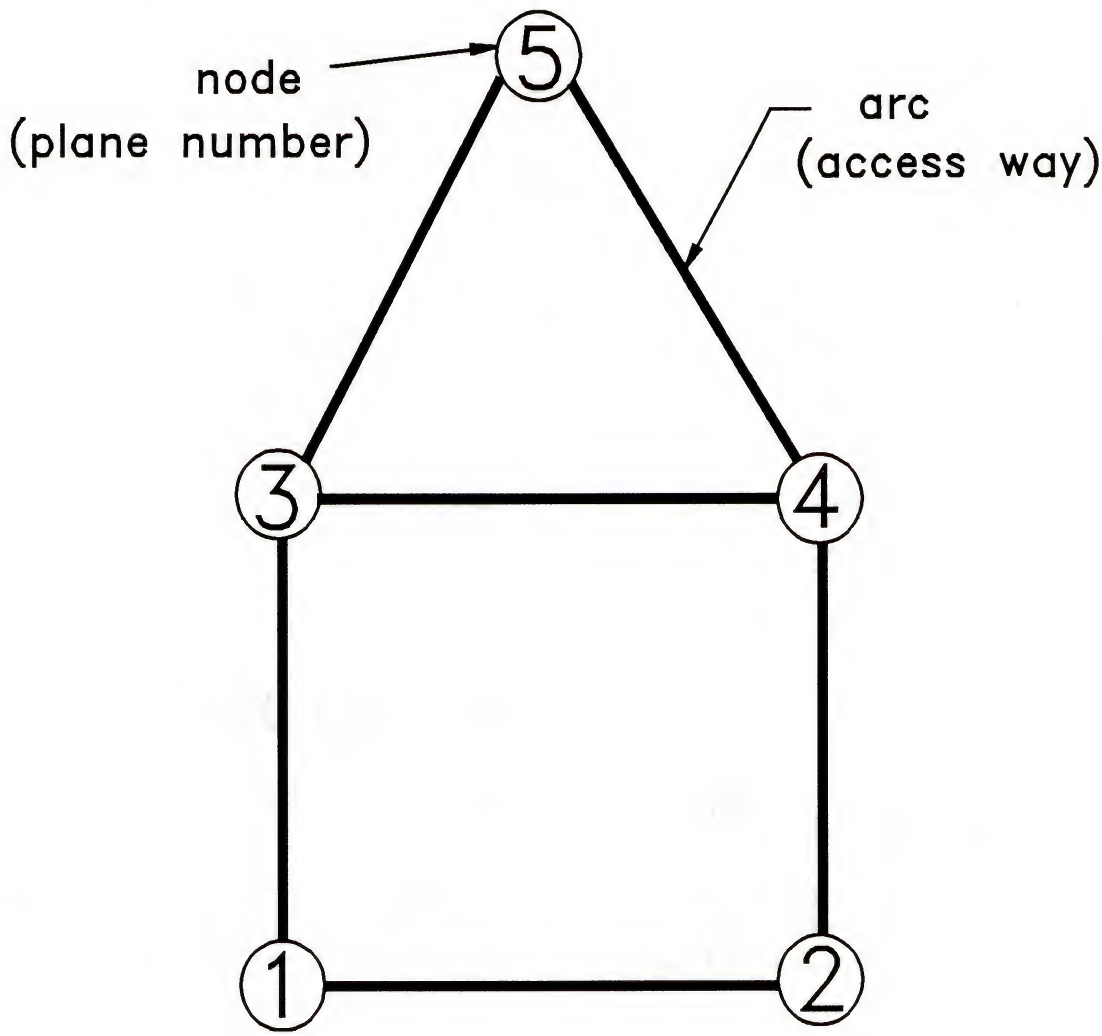


Figure 4.3 Graph for Plane Information in Figure 4.2

$$
\begin{array}{cc}
 & \text{plane no.} \\
\text{plane no.} &
\begin{array}{c|c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\hline
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1
\end{pmatrix}
\end{array}
\end{array}
\longrightarrow
\begin{array}{c}
\text{plane no.} \\
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{pmatrix}
0 & 26363 & 30291 & \infty & \infty \\
26363 & 0 & \infty & 26363 & \infty \\
30291 & \infty & 0 & 27636 & 30291 \\
\infty & 26363 & 27636 & 0 & 24545 \\
\infty & \infty & 30291 & 24545 & 0
\end{pmatrix}
\end{array}
$$

Adjacency Matrix

Cost Matrix

$$
\begin{array}{cc}
\text{plane no.} &
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{pmatrix}
0 & 0 & 0 & 2 & 3 \\
0 & 0 & 4 & 0 & 4 \\
0 & 4 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
3 & 4 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Via Matrix

Figure 4.4 Vertical Planning

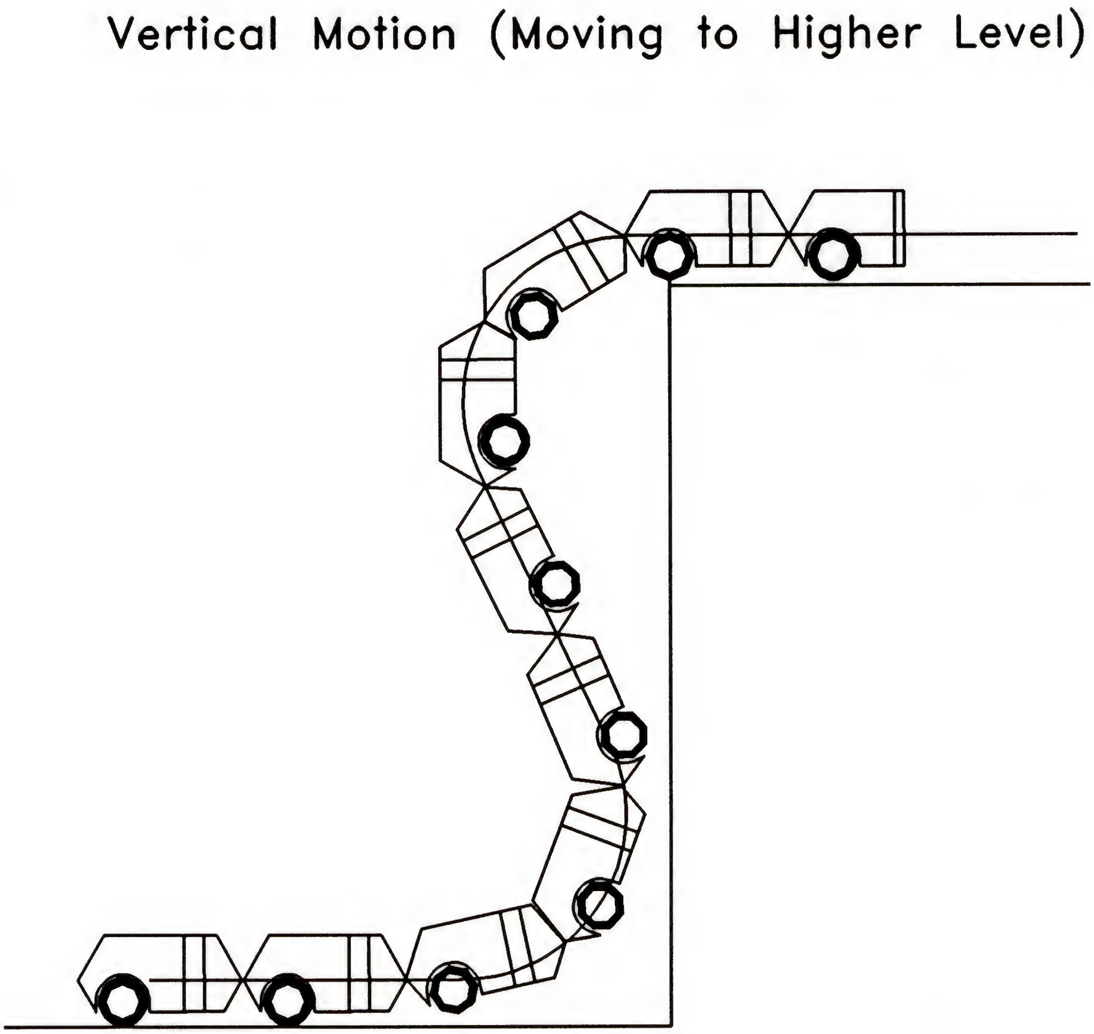


Figure 4.5 Vertical Motion Between Planes

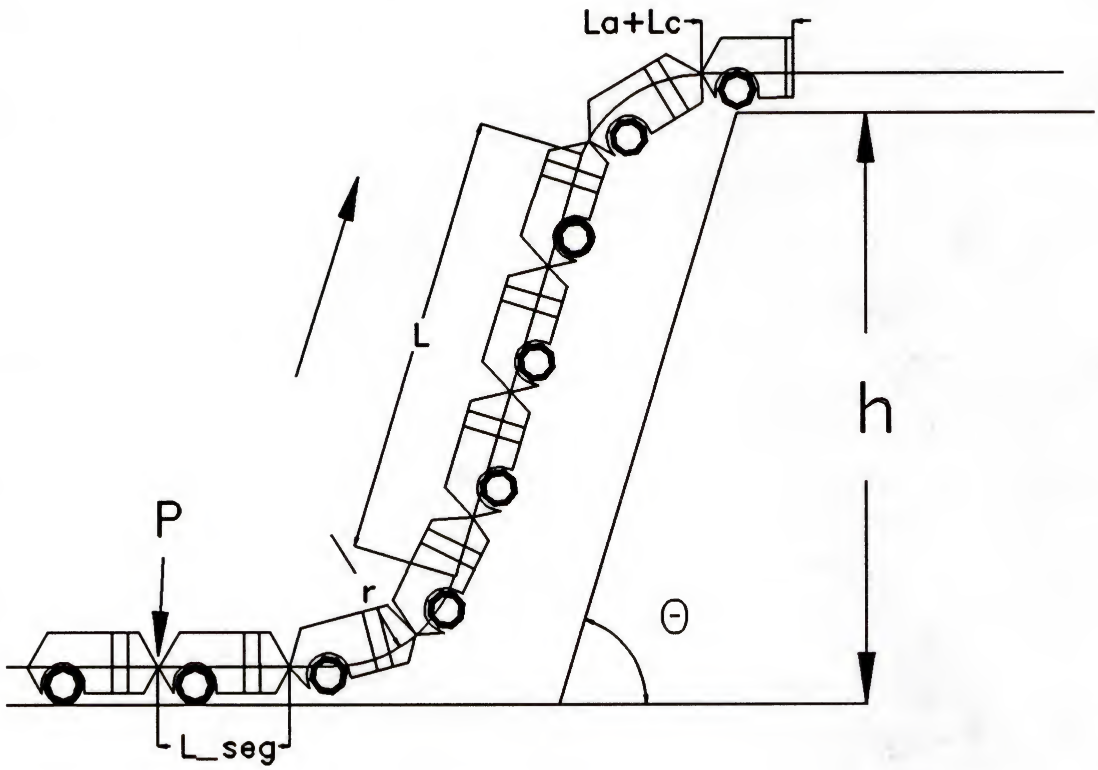


Figure 4.6 Vertical Motion to Higher Level

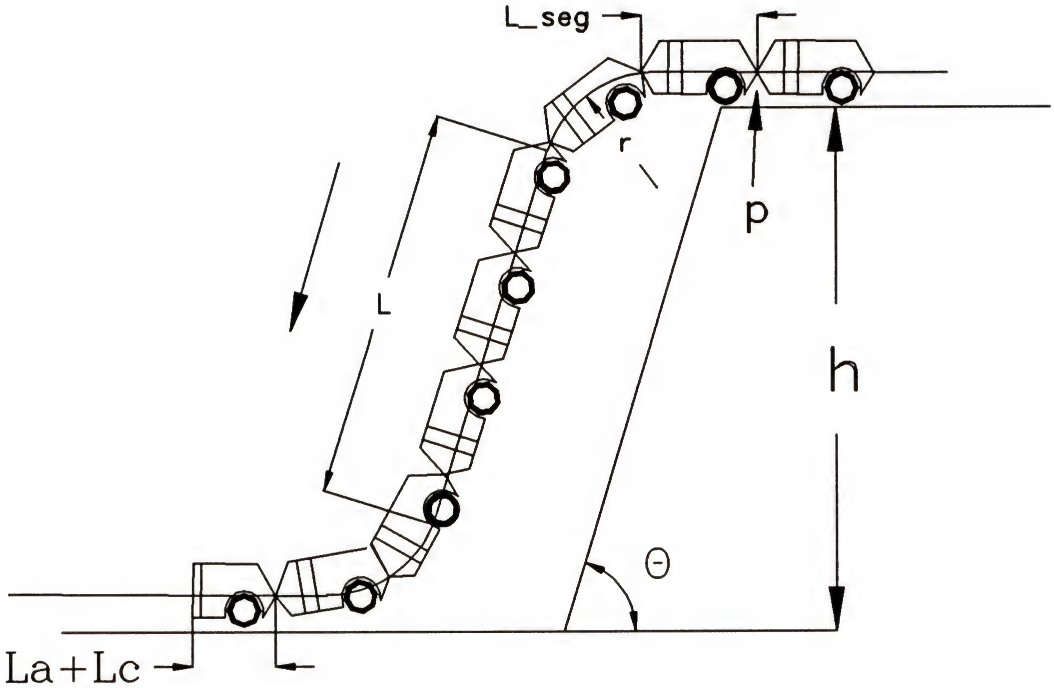


Figure 4.7 Vertical Motion to Lower Level

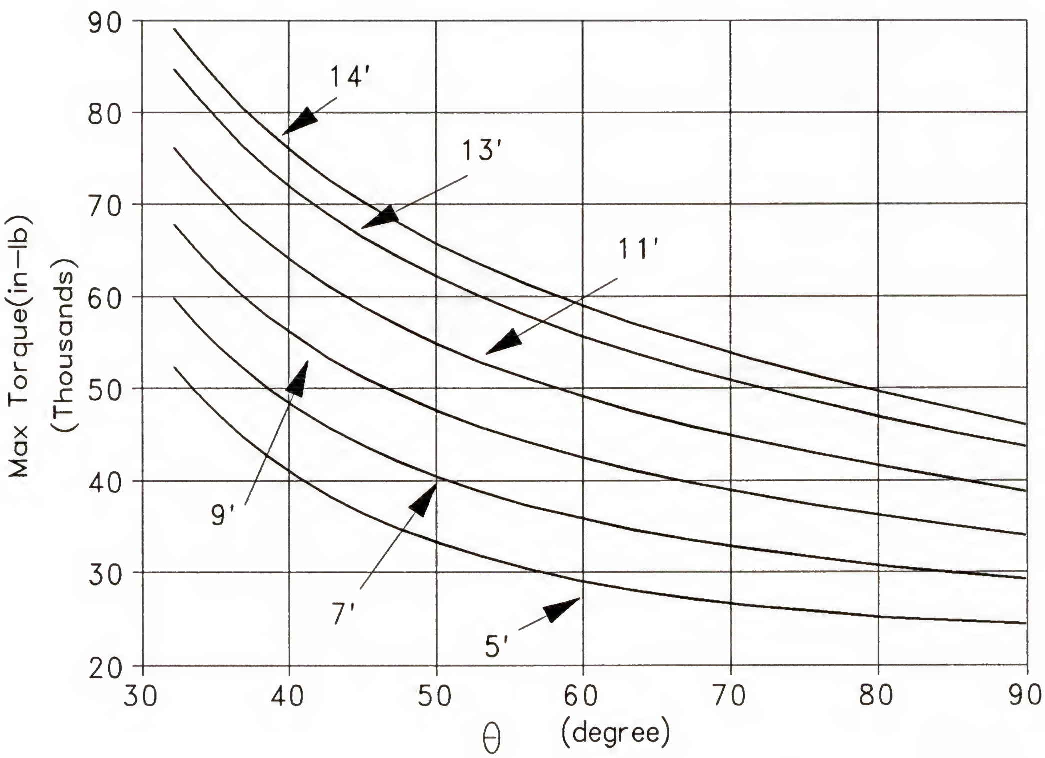


Figure 4.8 Maximum Torque on Joint
(For h = 5',7',9',11',13',14')

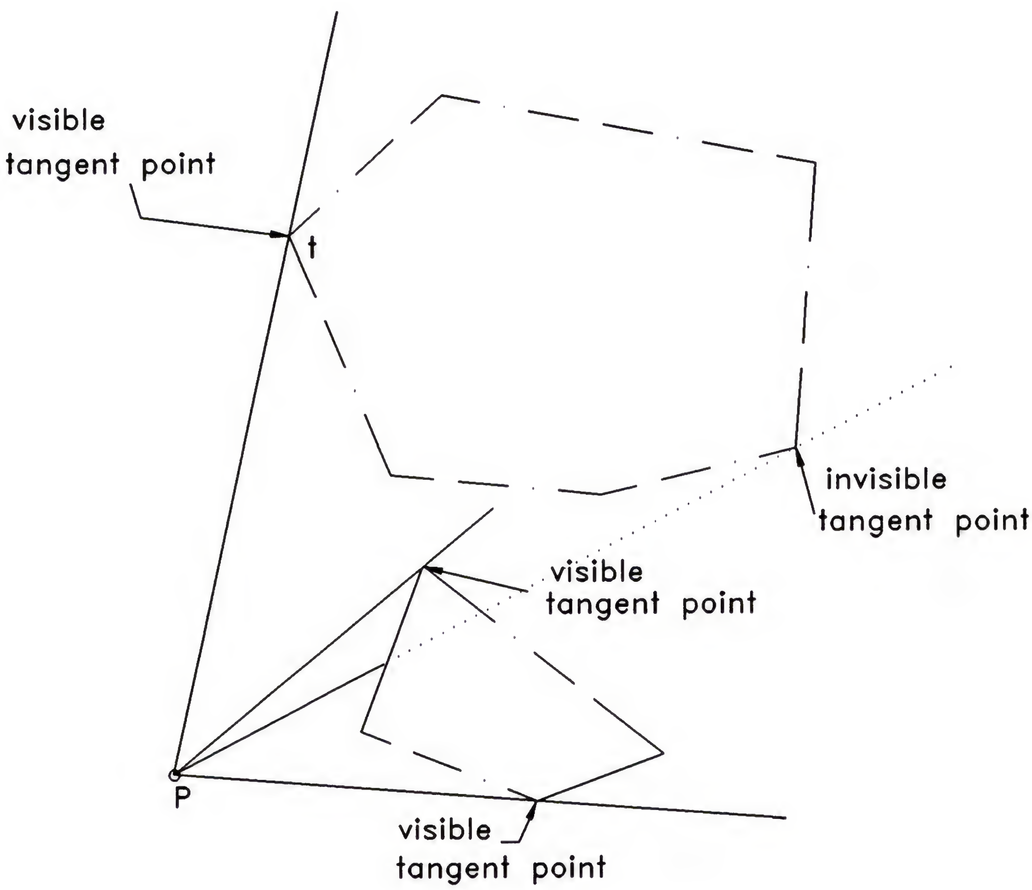


Figure 4.9 Tangent Point of the Polygon

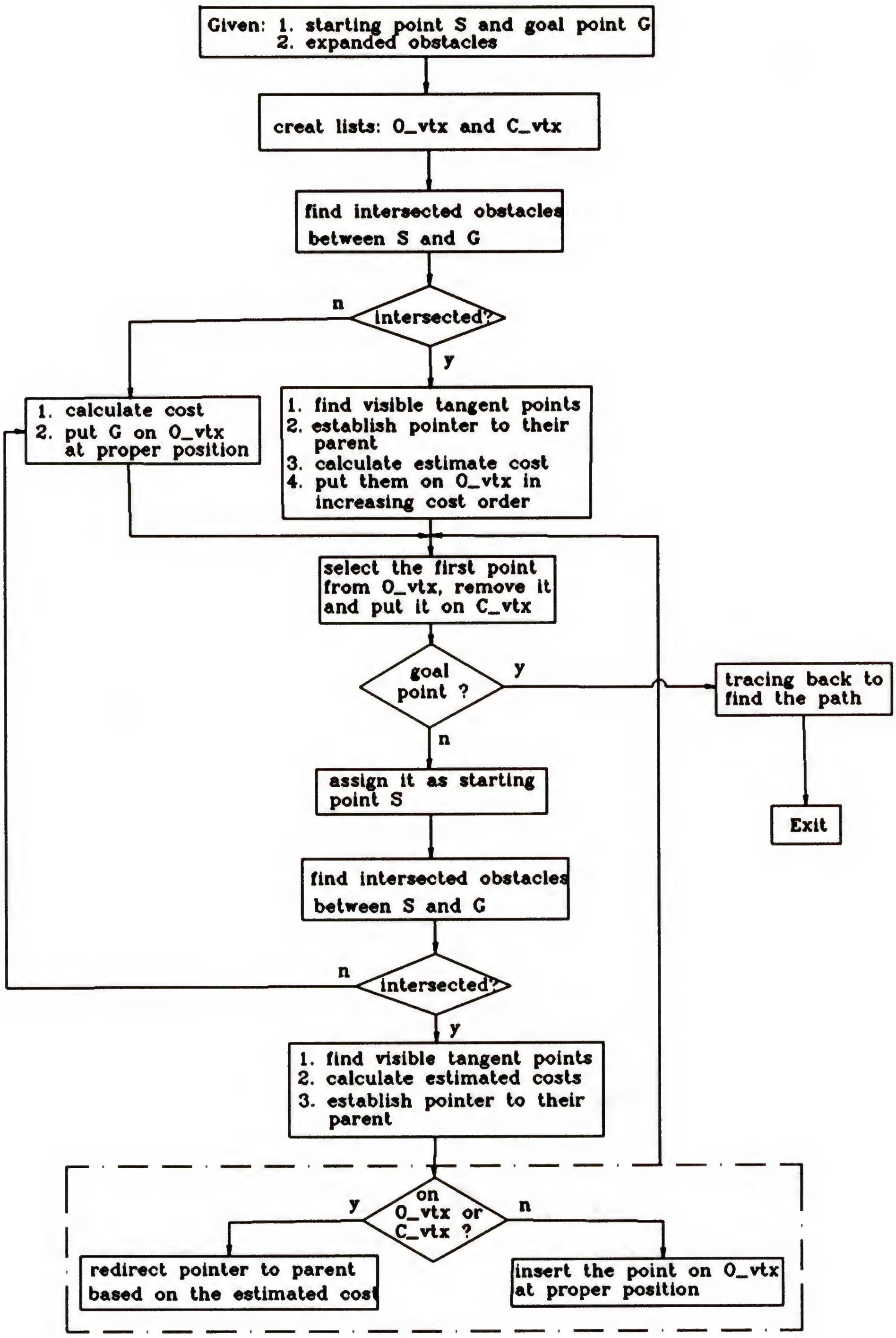


Figure 4.10 Diagram of Planning the Horizontal Path

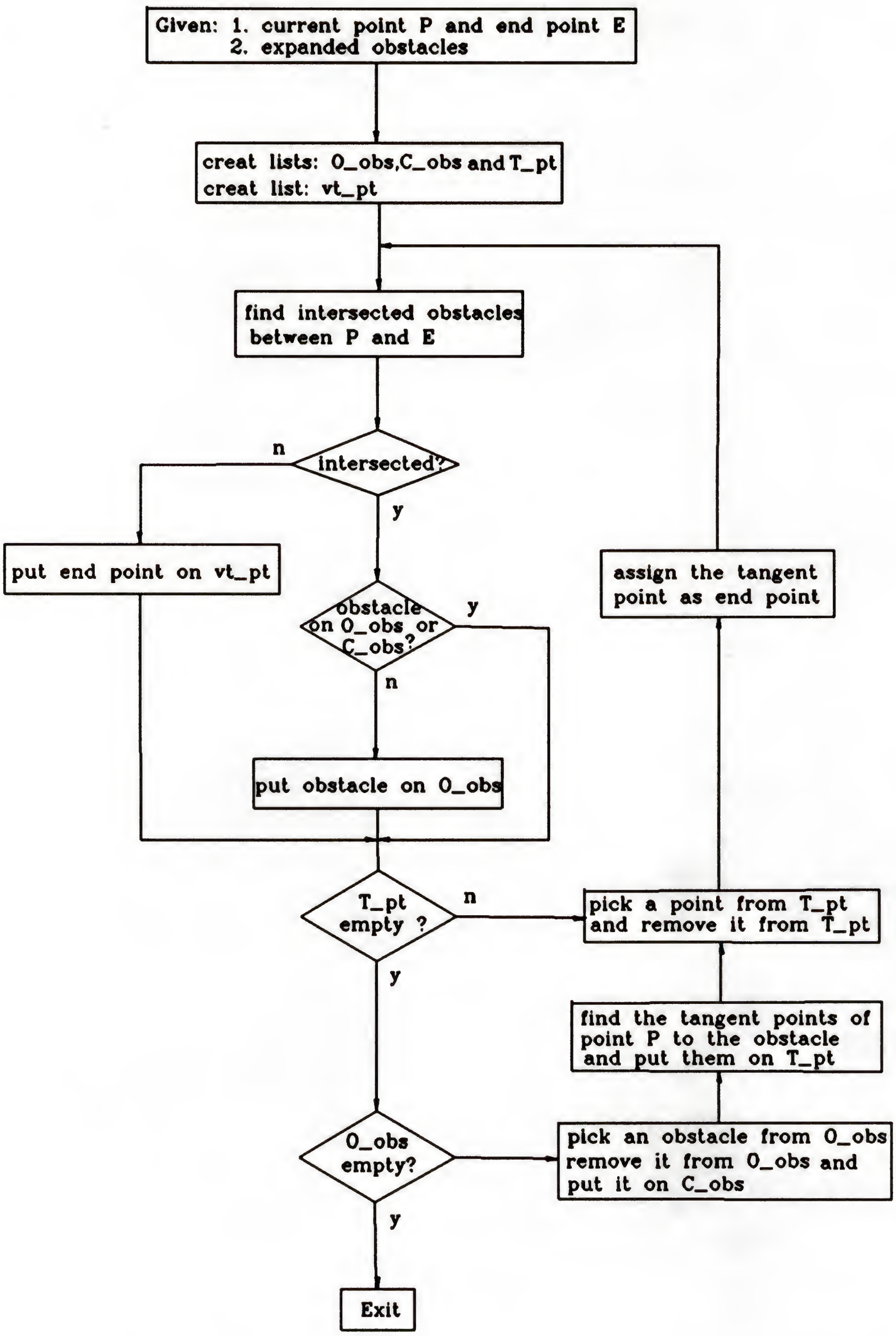


Figure 4.11 Procedures of Finding Visible Tangent Points

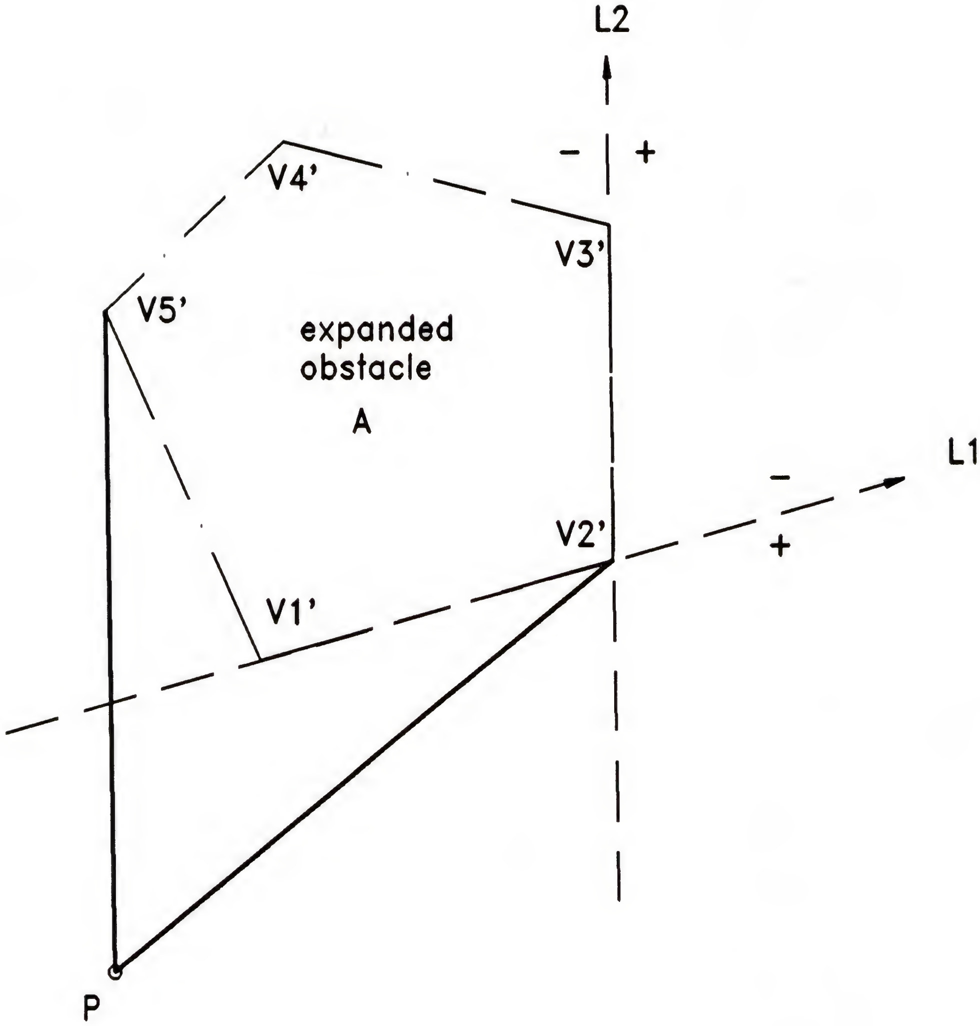


Figure 4.12 Find the Tangent Points

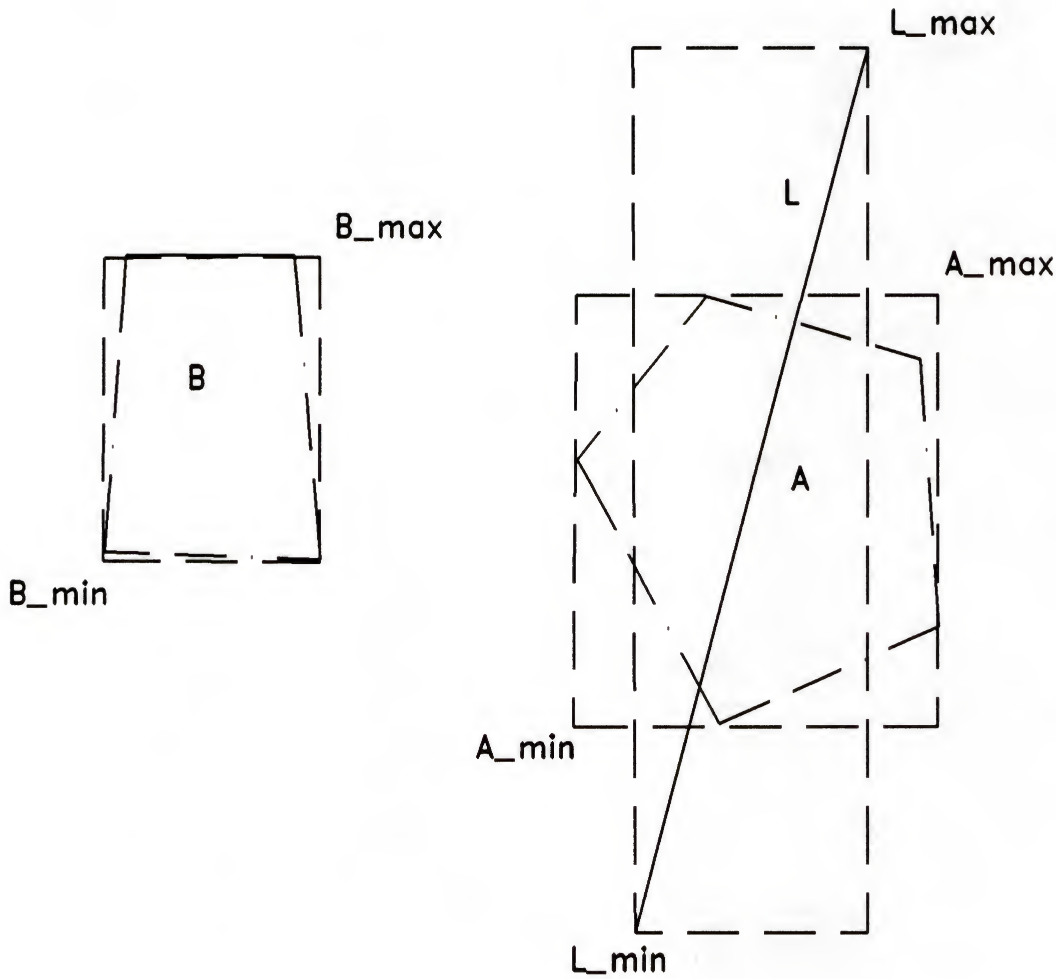


Figure 4.13 First Step of Intersection Check

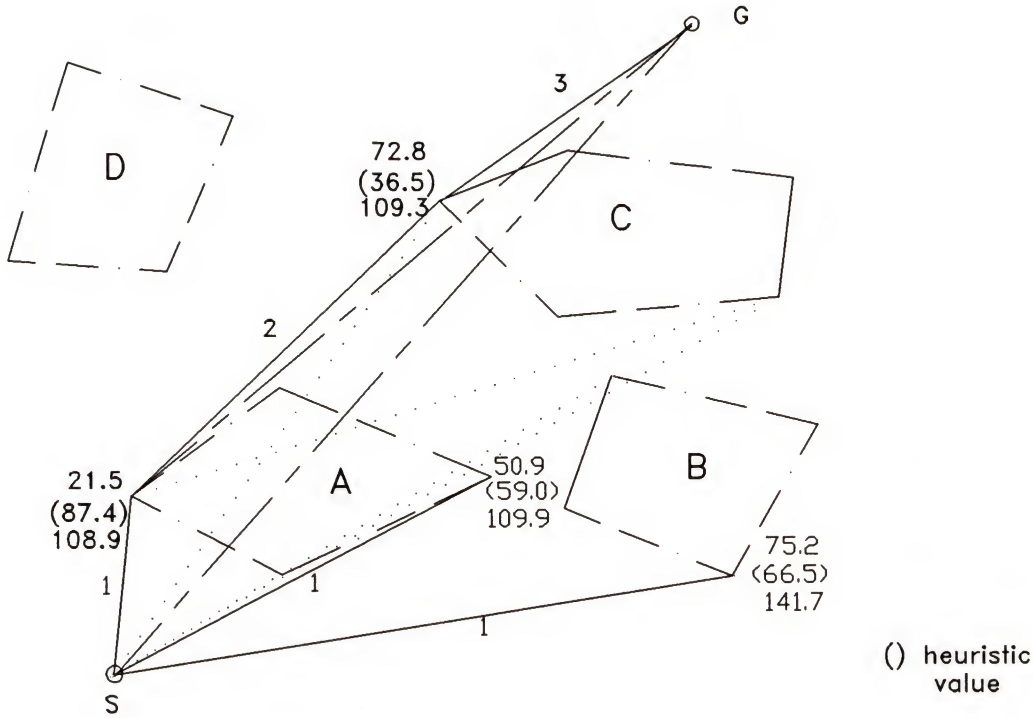


Figure 4.14 Horizontal Planning

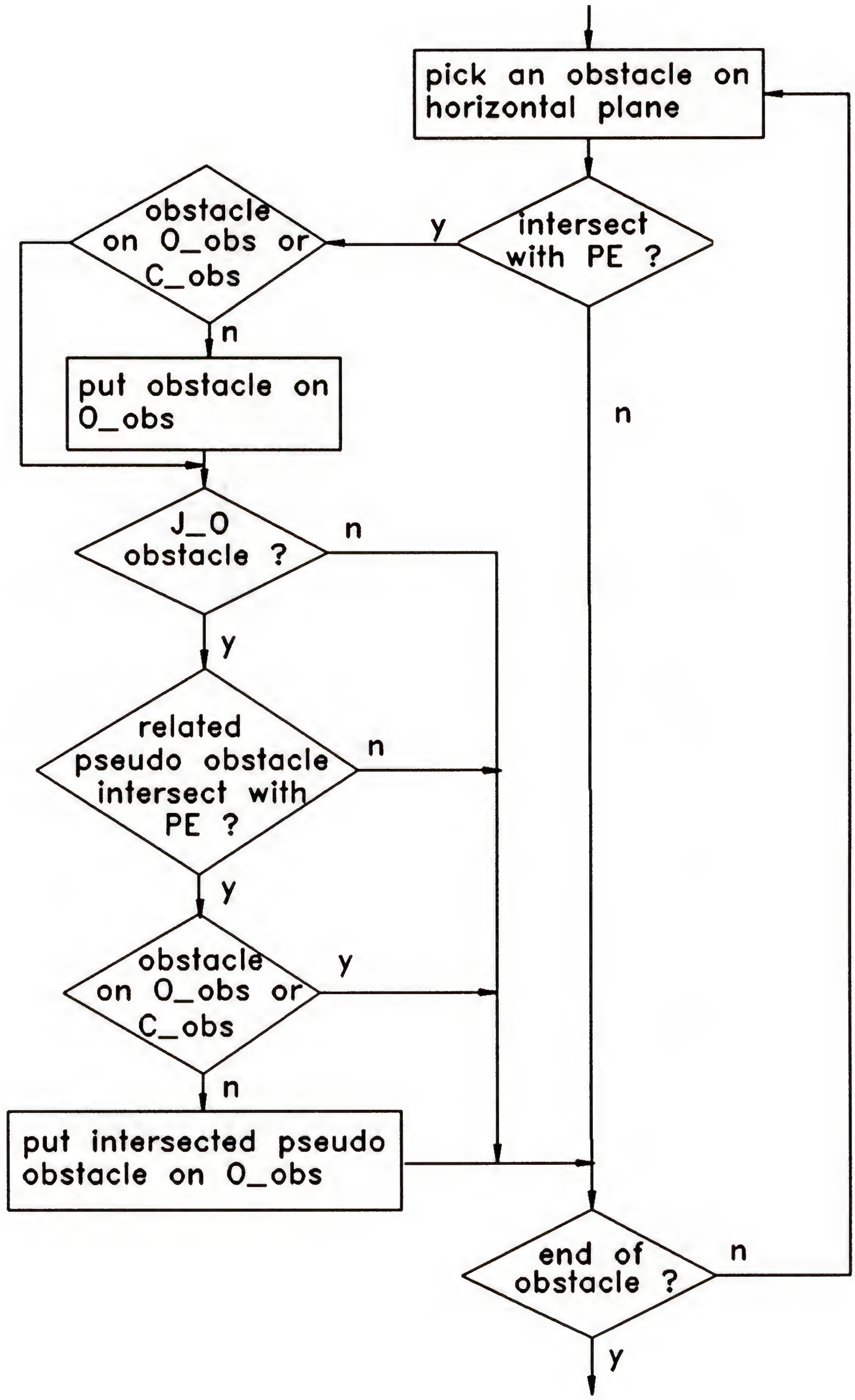


Figure 4.15 Criterion (a) for Planning a Vertical Path

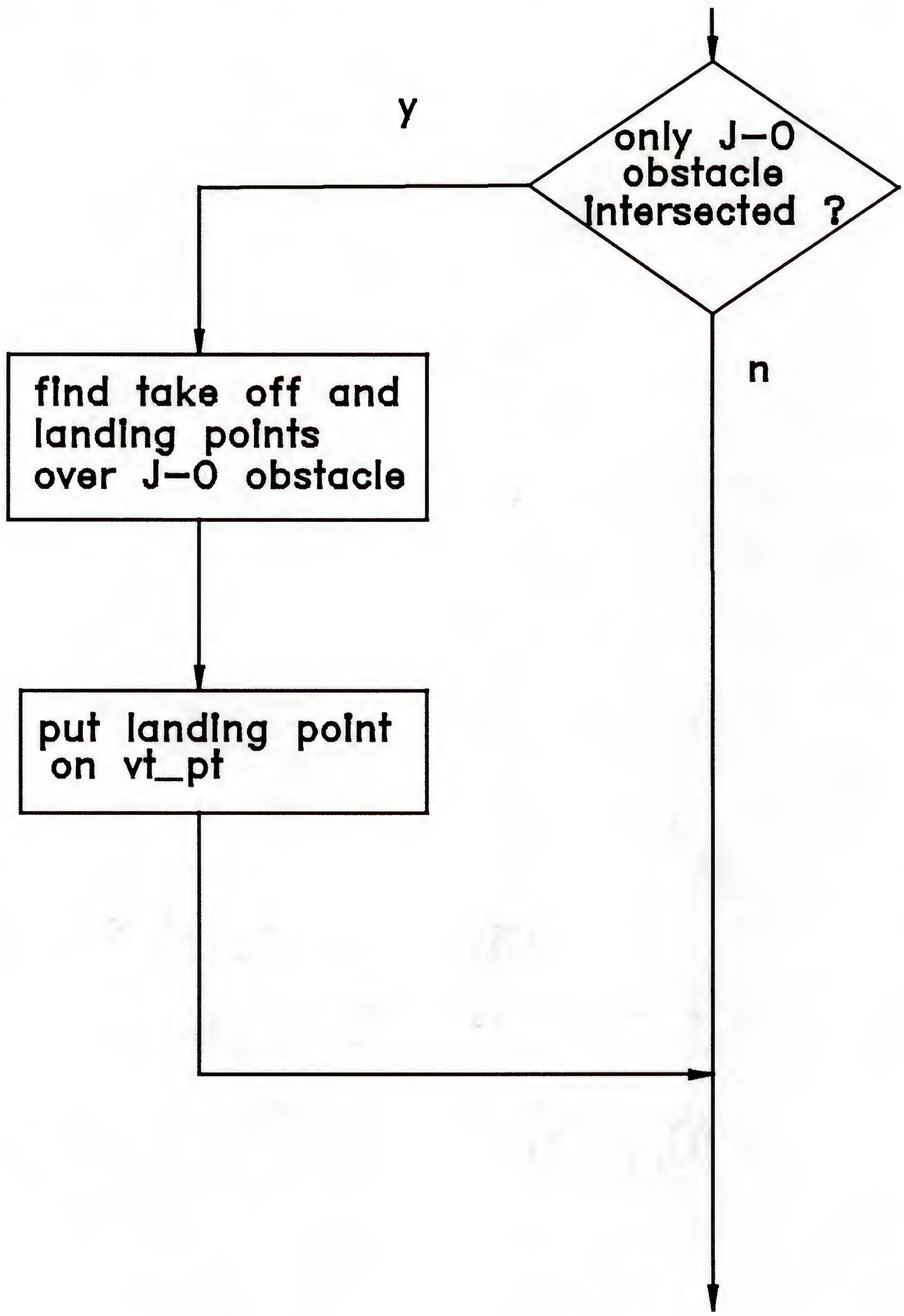


Figure 4.16 Criterion (b) for Planning a Vertical Path

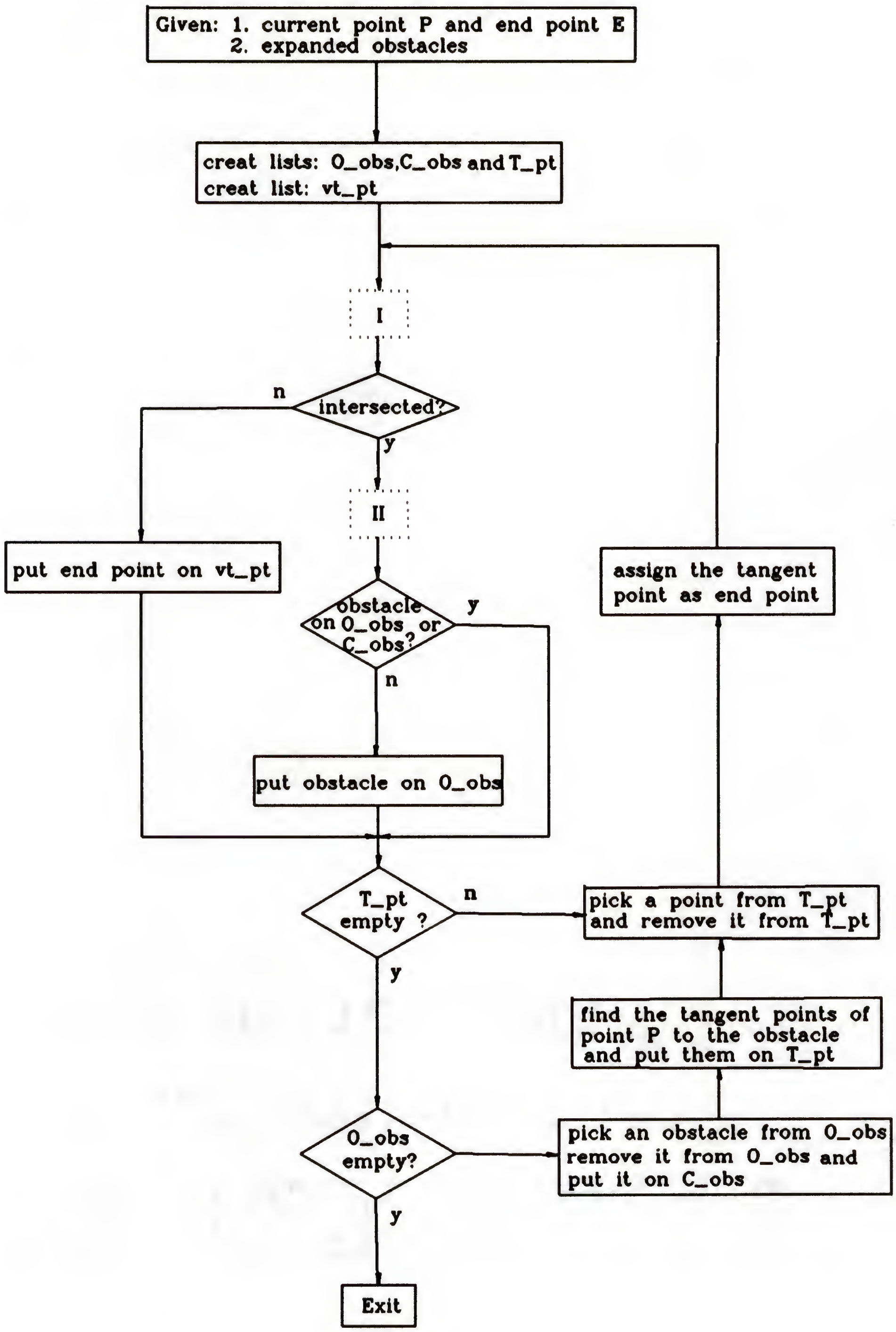


Figure 4.17 Diagram for Finding the Successive Points

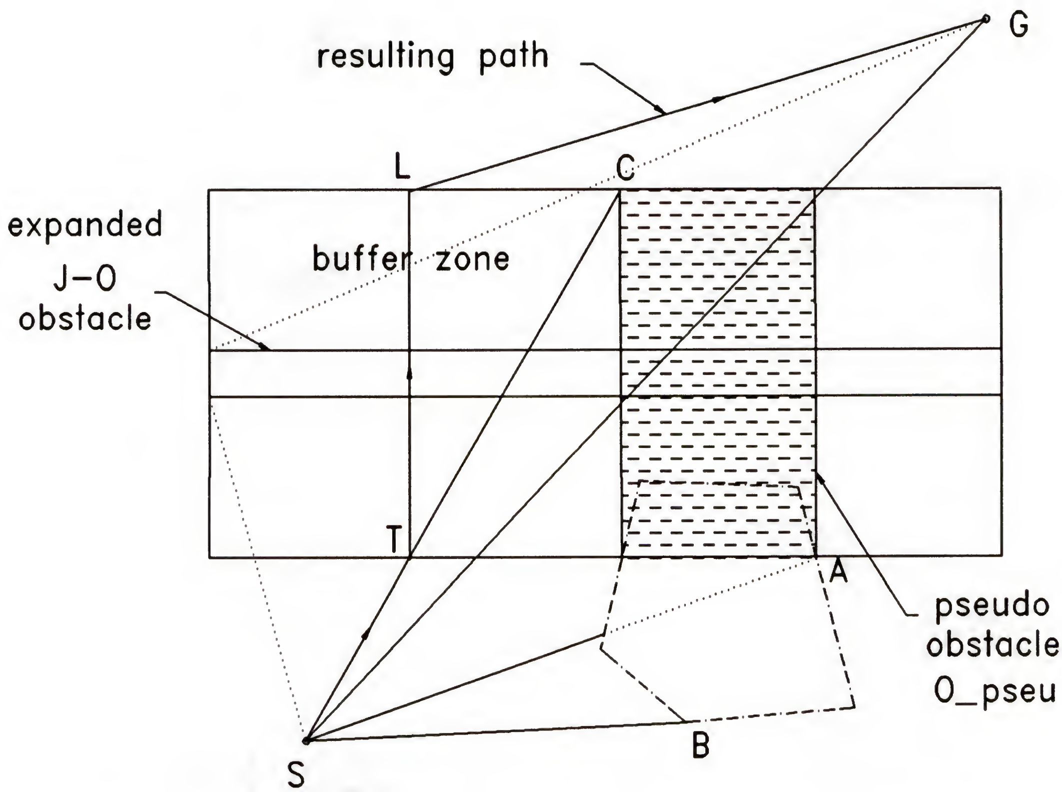


Figure 4.18 Planning a Proposed Vertical Path

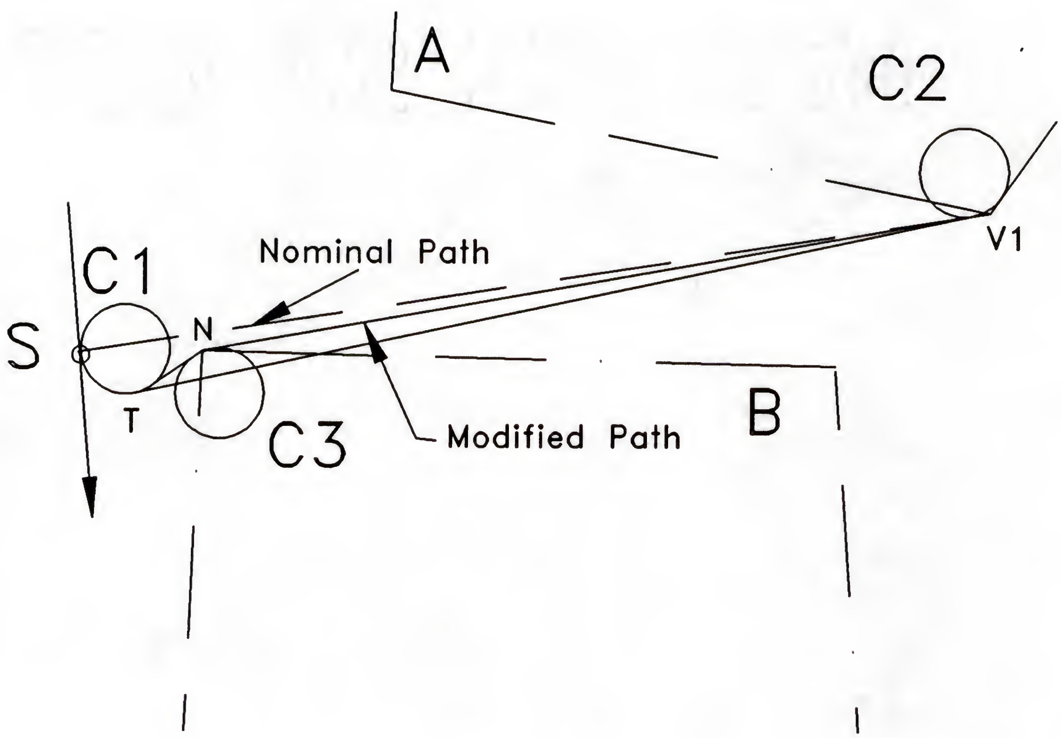


Figure 4.19 Planning for the Orientation

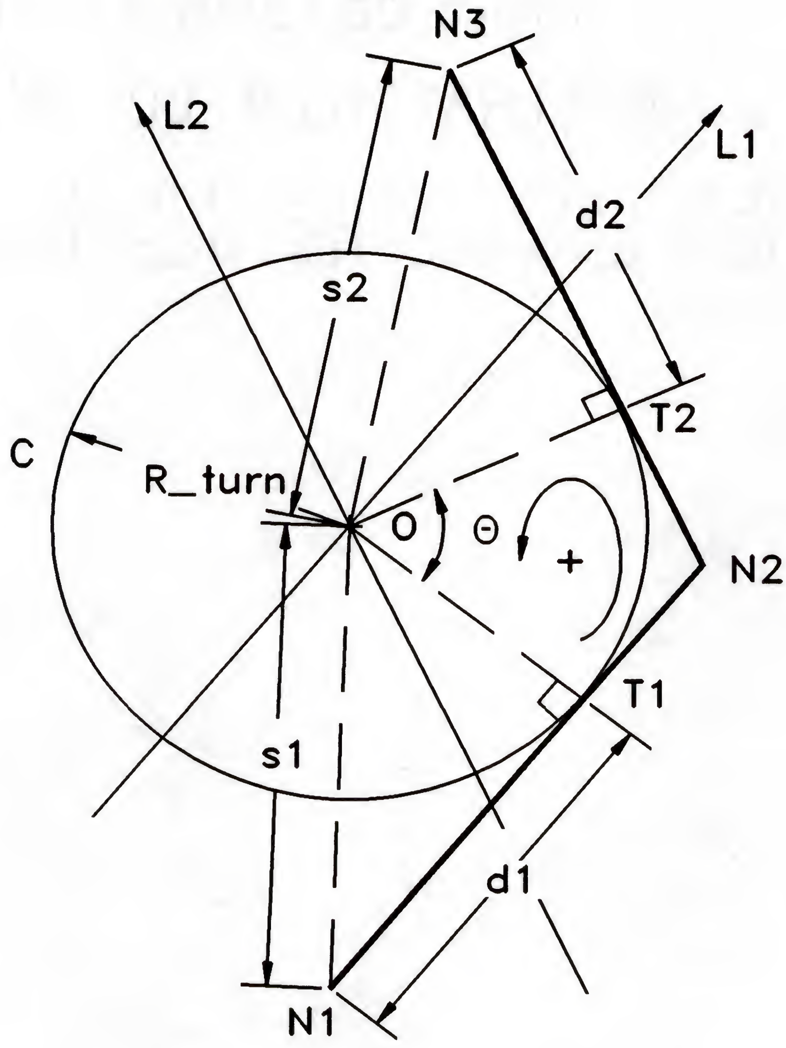


Figure 4.20 Modify Nominal Horizontal Path

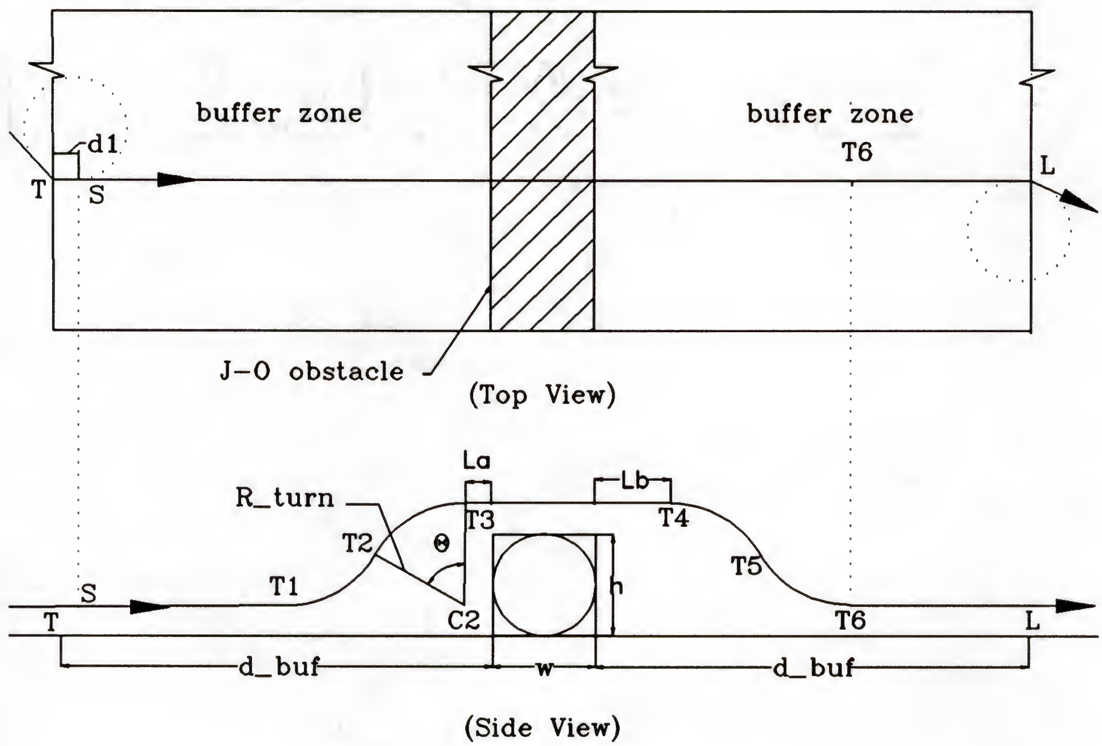


Figure 4.21 Planning Vertical Path Over J-O Obstacle

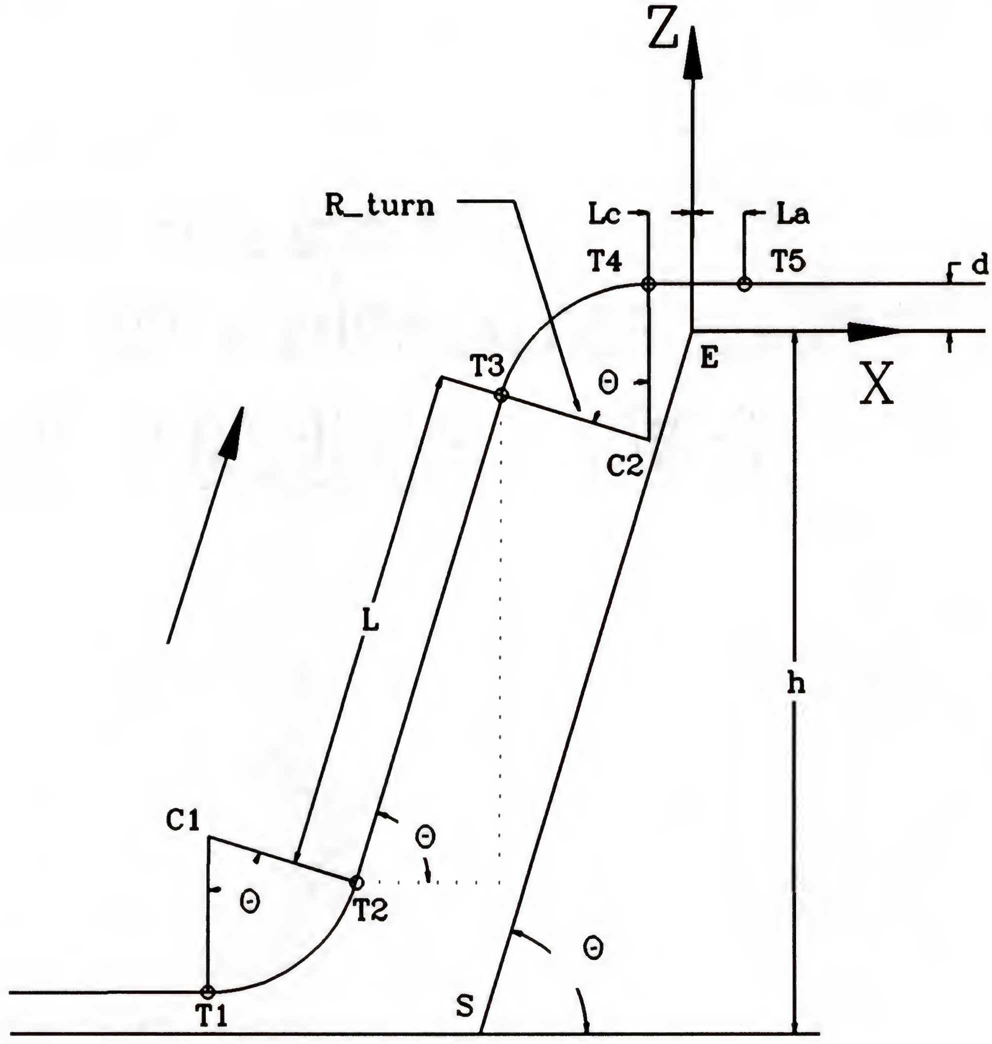


Figure 4.22 Vertical Path for Moving to a Higher Level

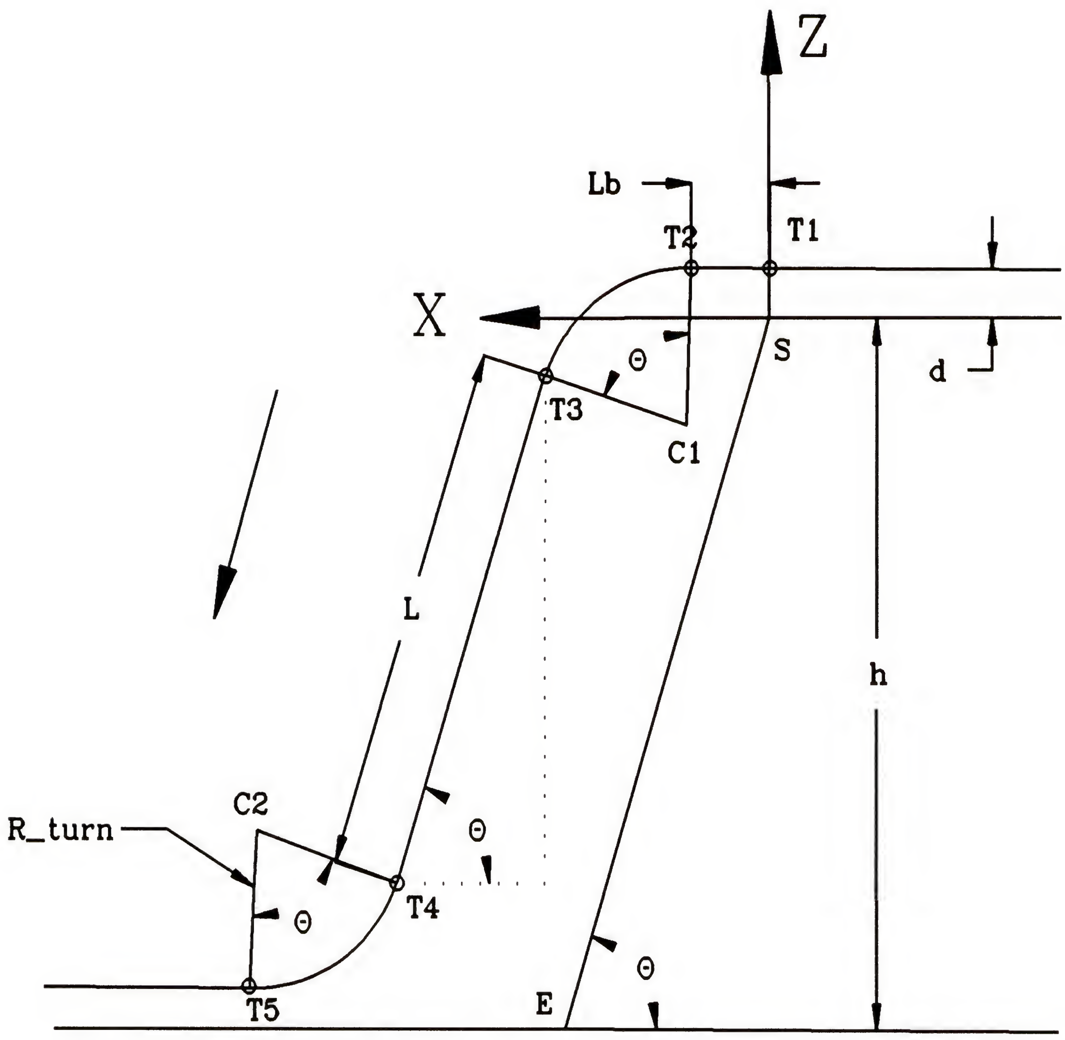


Figure 4.23 Vertical Path for Moving to a Lower Level

CHAPTER 5
DATA STRUCTURE AND COMPUTER GRAPHICS SIMULATION OF THE ATMS

5.1 Data Structure of Representing the Environment

The environment of the ATMS is made up of several planes in different elevations. On each of the horizontal planes, there are different types of obstacles. Access ways are defined between horizontal planes. Basically, three kinds of data lists are used to store different information about the environment: plane list, obstacle list, and vertex list. The environment is defined by an array of plane lists, a linearly linked obstacle list, and a circular linked vertex lists. The data structure representing the environment is shown in Figure 5.1.

The data which are contained on the plane list are stated as follows:

1. the elevation of the plane,
2. access ways,
3. pointer to the root obstacle on the plane.

For each of the horizontal planes in the environment, a plane list is created to store the information related to the plane. In item 2, the coordinates of the starting and ending points of the access ways and the from and to planes of the access

ways are stored. As shown in Figure 5.1, the obstacle lists for the obstacles on the horizontal plane are linearly linked. To find the obstacles on the horizontal plane, only a pointer to the root of the linked obstacle list is needed, and the pointer is stored in item 3 of the plane list.

The data which are contained on an obstacle list are stated as follows:

1. obstacle type,
2. number of vertex of the obstacle,
3. height,
4. pointer to the root vertex of the obstacle,
5. pointer to the corresponding expanded obstacle,
6. pointer to the root pseudo obstacle (J-O obstacle only),
7. pointer to the buffer zone (J-O obstacle only),
8. pointer to the next obstacle.

The first item on the obstacle list is the type of the obstacle, which is used to distinguish J-O obstacles from other types of obstacles. A pointer to its expanded obstacle is stored in item 5. If the obstacle is a J-O obstacle, pointers to the related pseudo obstacles and buffer zone are stored in items 6 and 7. The data structure for a J-O obstacle is shown in Figure 5.2. The pointer which is stored in item 8 links the obstacles on the horizontal plane.

Three sub-lists are created to store the data about the expanded obstacle, pseudo obstacle, and buffer zone. They are stated as follows:

<u>Expanded obstacle list</u>

1. number of vertex of the expanded obstacle,
2. maximum and minimum coordinates,
3. pointer to the root vertex of the expanded obstacle.

<u>Pseudo obstacle list</u>

1. maximum and minimum coordinates,
2. pointer to the root vertex of the pseudo obstacle,
3. pointer to the next pseudo obstacle.

<u>Buffer zone list</u>

1. maximum and minimum coordinates,
2. pointer to the root vertex of the buffer zone.

The vertex list is created to store the data about the vertices of an obstacle. The data which are contained on the vertex list are stated as follows:

1. x, y, z coordinates,
2. coefficients of the line equation,
3. pointer to the next vertex,
4. pointer to the previous vertex.

Item 1 of the vertex list stores the coordinates of the vertex. Item 2 stores the data of the edge of the obstacle. The data is the coefficients of the line equation of the line connecting the current vertex to the next vertex of the obstacle. Items 3 and 4 store the pointers to the next and previous vertices and creates a circular linked lists of the vertex, and the vertices of the obstacle are linked in counter clockwise order.

5.2 Linked Representation of Search Tree

In the horizontal planning, a search tree is developed while the graph search proceeds. For each node in the tree, a treenode list is allocated to store the information about

the node and to establish the relationship with other nodes in the tree. The treenode list is displayed as follows:

1. x, y, z coordinates,
2. node type,
3. height,
4. actual cost,
5. total cost,
6. pointer to its father node,
7. pointer to its brother node,
8. pointer to next node,
9. pointer to previous node.

Item 1 stores the position of the node, which is the position of a vertex of the obstacle in the environment. Item 2 stores the information about the type of the node, this information helps to plan a proper path (horizontal or vertical path) in the operative planning. Item 3 stores the height of the J-O obstacle, which is used to plan a vertical path over the J-O obstacle. Items 4 and 5 store the costs calculated by the cost function. The total cost is used to decide the node for next expansion. The pointers in items 6 and 7 establish the relationship of the node in the search tree (see Figure 5.3). After the goal point is reached, the pointer which is stored in item 6 helps to trace back to the starting point from the goal point and create the nominal horizontal path. Pointers which are stored in item 8 and 9 are used to create a double linked lists for the nodes in the list O_vtx in Figure 4.10. The list O_vtx keeps all of the leaf nodes of the tree in a increasing cost order. After a node is expanded, the costs of its successors are estimated and the successors are inserted in the list O_vtx at the proper positions (Figure 5.4).

5.3 Simulation Results

The algorithms which have been developed in the previous chapters are implemented in C language on a Silicon Graphics 4D-310VGX workstation. The simulation results are displayed in two ways. In Section 5.3.1, examples of graph building are presented. The development of the search graph are shown in the series of figures. The simulation results of a planned path and the animated motion of the ATMS follows the planned path are shown in Section 5.3.2.

5.3.1 Examples of Graph Building

Two examples of the graph building which are created by the implemented path planning algorithms are presented. In these two examples, the starting and goal positions are specified by circles at S and G and the short line segment from the centers of the circles are the specified orientations.

In the first example (Figures 5.5.a - 5.5.f), the environment consists of three convex polygonal obstacles. Each succeeding figure in the series shows the expansion of the next chosen node in the search graph. It can be seen that the search does not span the entire graph and only one node which is not on the final path is expanded. Figure 5.5.e shows the modification of the path for the specified

orientations. The final collision free path is shown in Figure 5.5.f.

Figures 5.6.a - 5.6.g show the second example of graph building. In this example, a J-O obstacle is appeared in the environment of four obstacles. In series of expansions from Figure 5.6.a to 5.6.e, it can be seen that both vertical and horizontal paths are considered to be planned to pass the J-O obstacle in the horizontal planning. The final path is decided by the total cost of the path. In this example, the proposed vertical path over the J-O obstacle is the best choice. The final planned path is shown in Figure 5.6.g.

5.3.2 Computer Graphics Simulation of the ATMS

The final planned path and the animated motion of the ATMS are displayed on a Silicon Graphics 4D-310VGX workstation. A working environment which consists of more than 40 obstacles is designed to verify the result.

Following are the results which have been demonstrated by the computer graphics simulation for different situations:

1. Vertical path over a J-O obstacle

   Figures 5.7 shows the planned vertical path over the J-O obstacle. The ATMS successfully follows the planned path to jump over the J-O obstacle.

2. Final planned path is decided by the total cost

   In Figure 5.8, the position of the goal point is specified closer to the end of a J-O obstacle than the one in Figure 5.7, a horizontal path is then planned to pass the J-O obstacle.

3. Different path is planned for different joint torque capacity

   Figures 5.9 and 5.10 have the same starting and goal positions, by specifying different joint capacities, the system plans different via planes.

4. Vertical path between planes

   Figure 5.11 shows the planned vertical path and the motion of the ATMS to move to a lower plane.

5. Horizontal collision-free path

   In Figure 5.12, a horizontal collision-free path is planned through a labyrinth entrance.

6. Collision-free path and continuous movement

   Figures 5.13 and 5.14 show that the ATMS continuously travels to the second goal point after the first goal point has been reached.

From the results of computer graphic simulation, it is shown that the algorithms which have been developed in the previous chapters are capable of planning a three-dimensional collision-free path in a given working environment. The ATMS can also follow the planned path by vertical and horizontal motions.

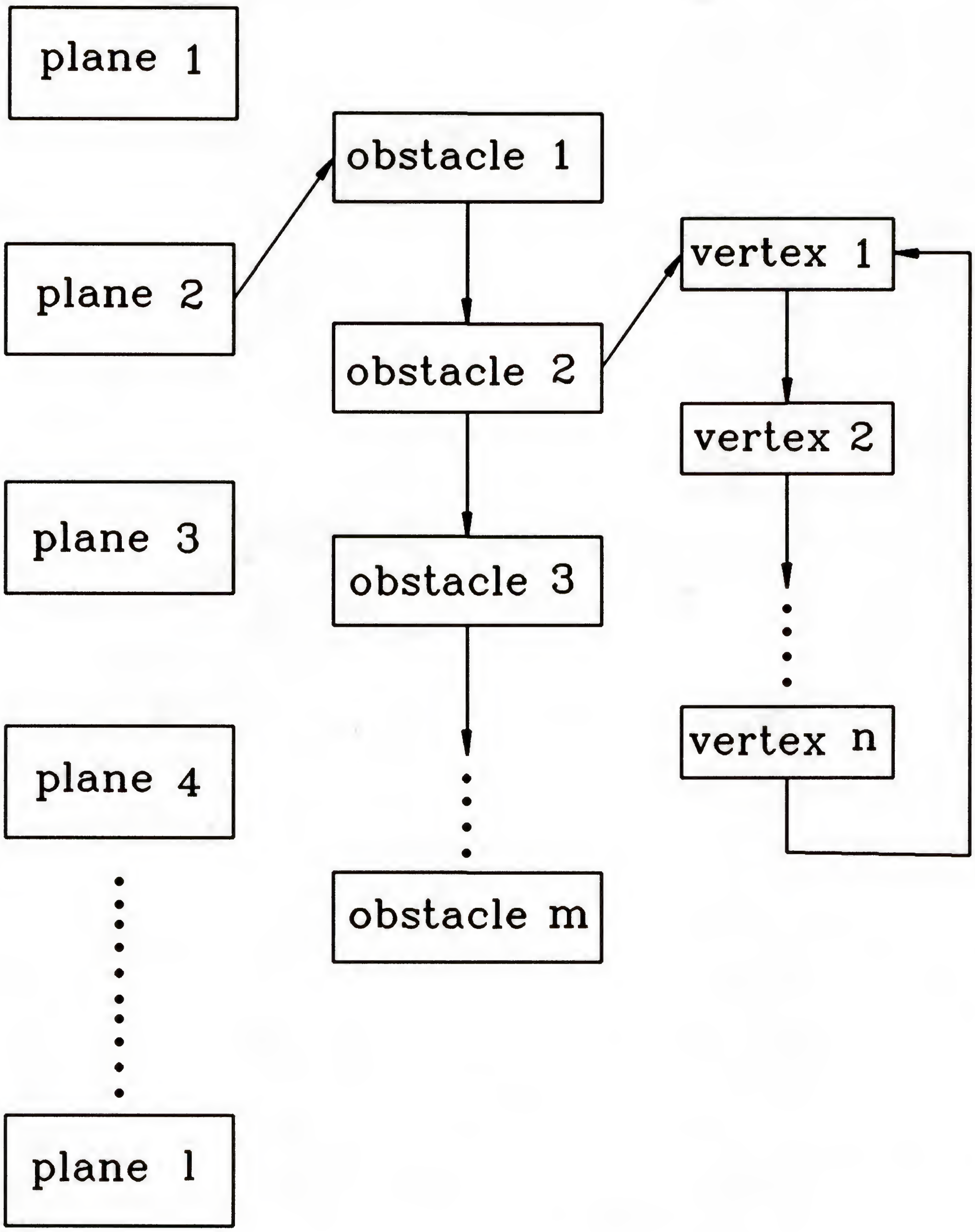


Figure 5.1 Data Structure for Representing the Environment

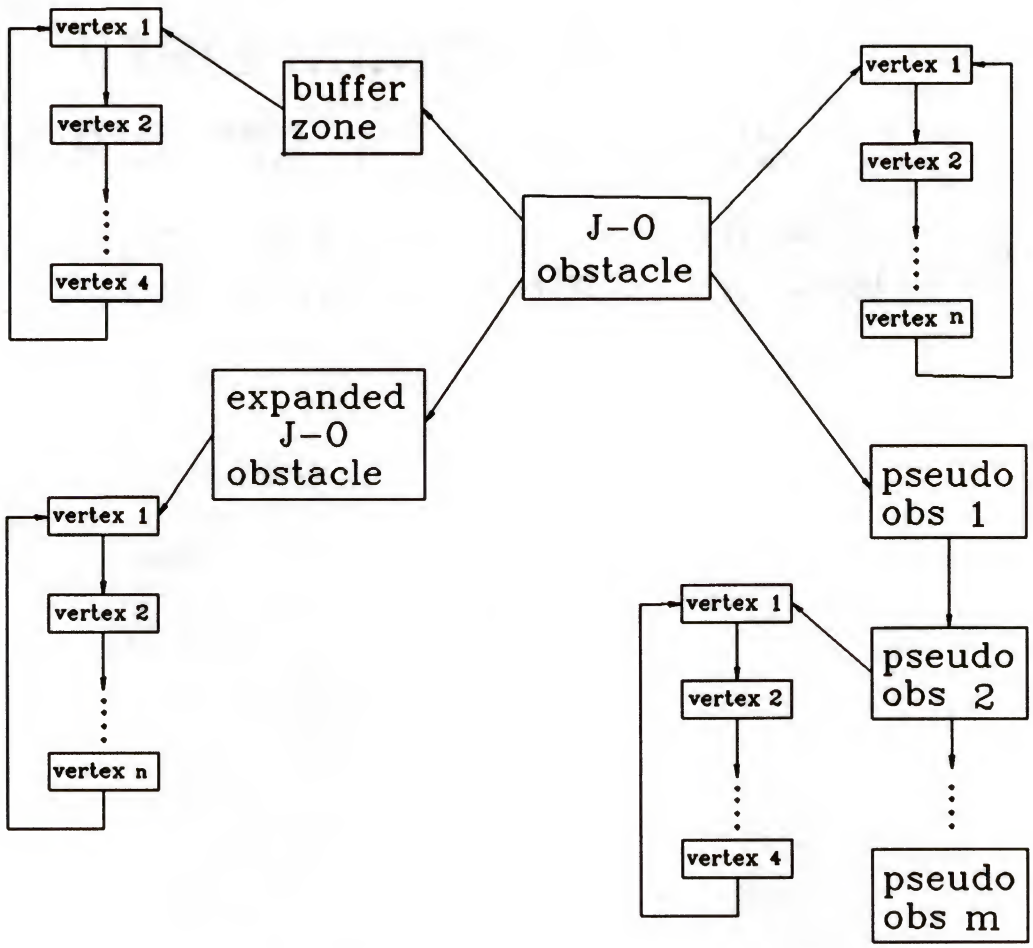
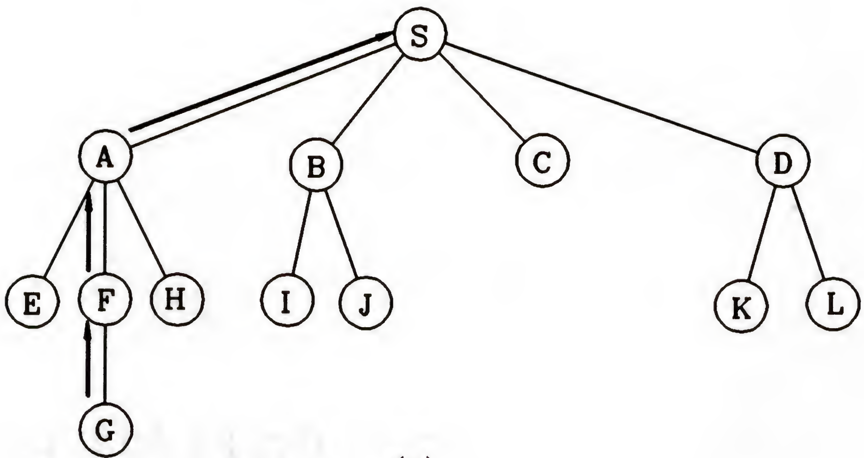
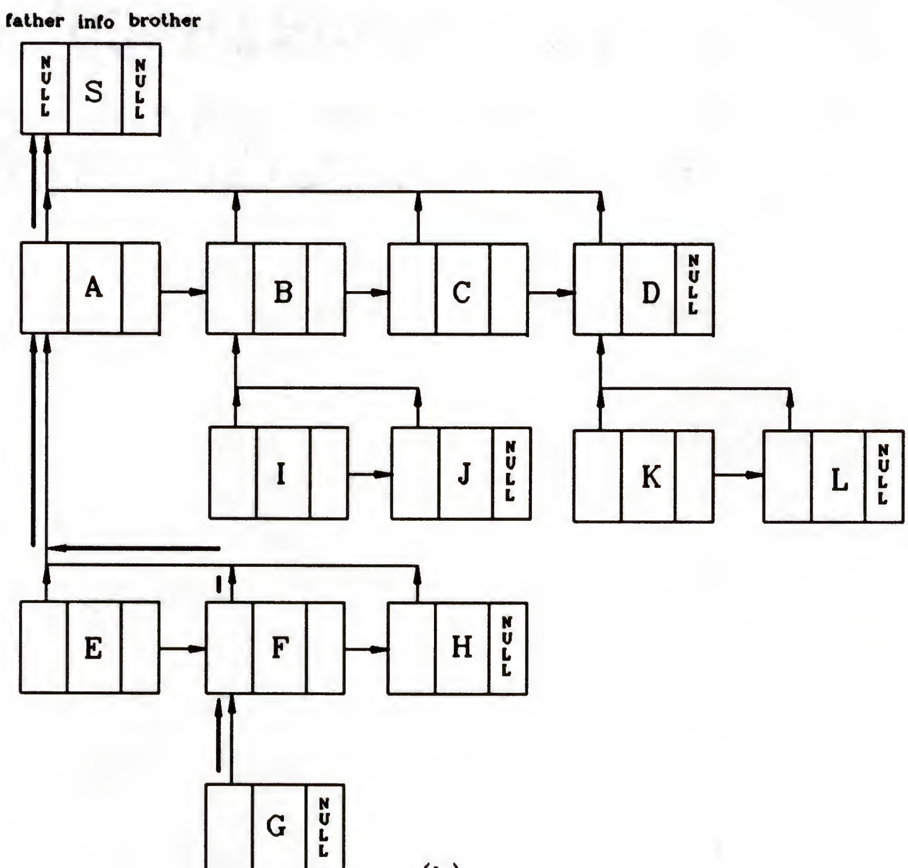


Figure 5.2 Data Structure for Representing a J-O obstacle



(a)



(b)

Figure 5.3 Linked Representation of Tree

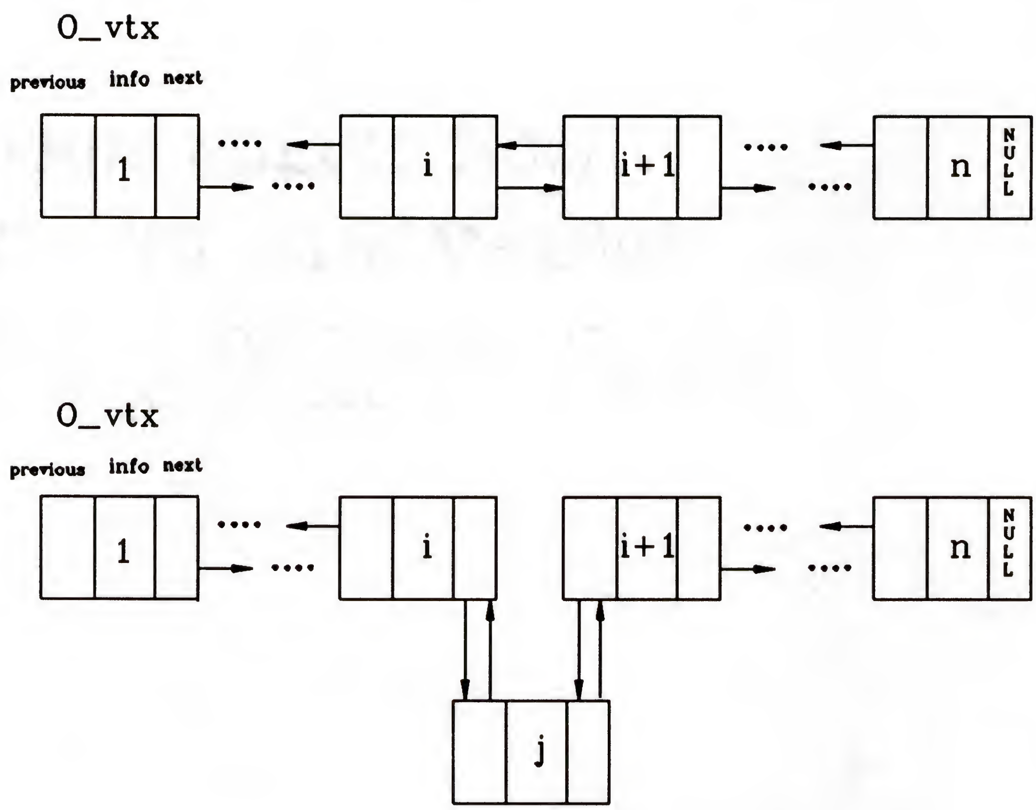


Figure 5.4 Insert a Node in O_vtx List

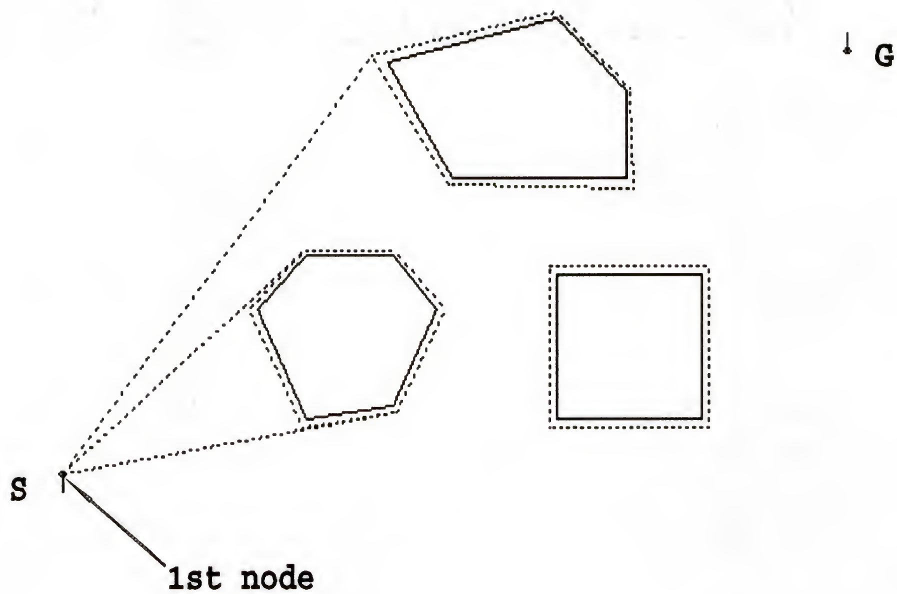


Figure 5.5.a Expansion of 1st Node

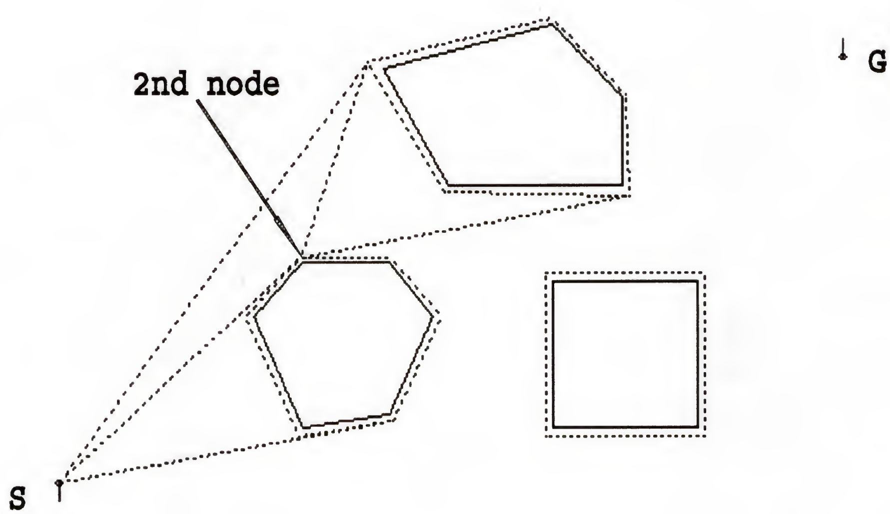


Figure 5.5.b Expansion of 2nd Node

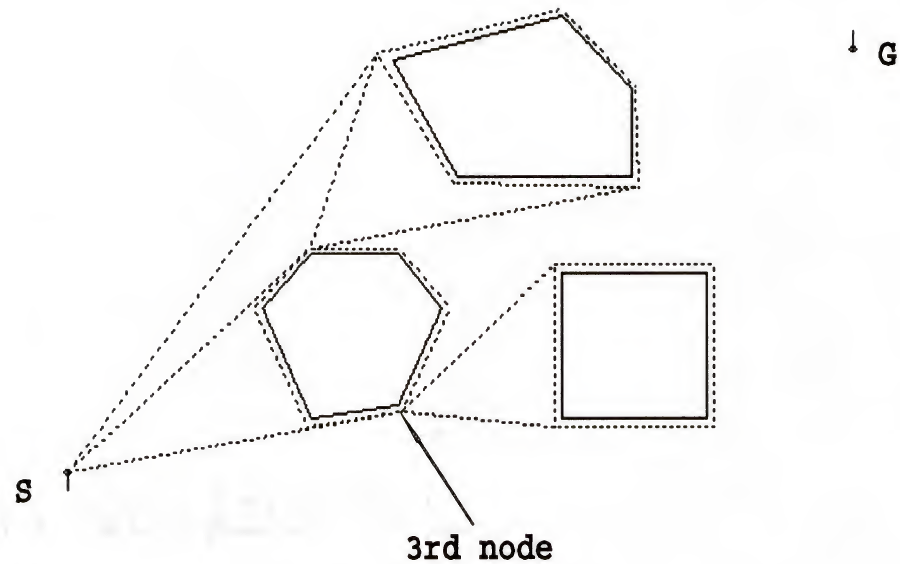


Figure 5.5.c Expansion of 3rd Node

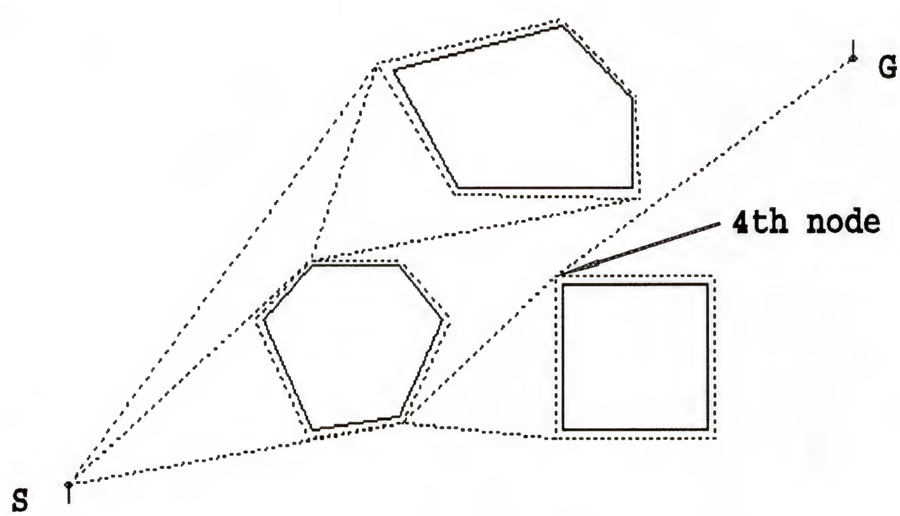


Figure 5.5.d Expansion 4th Node

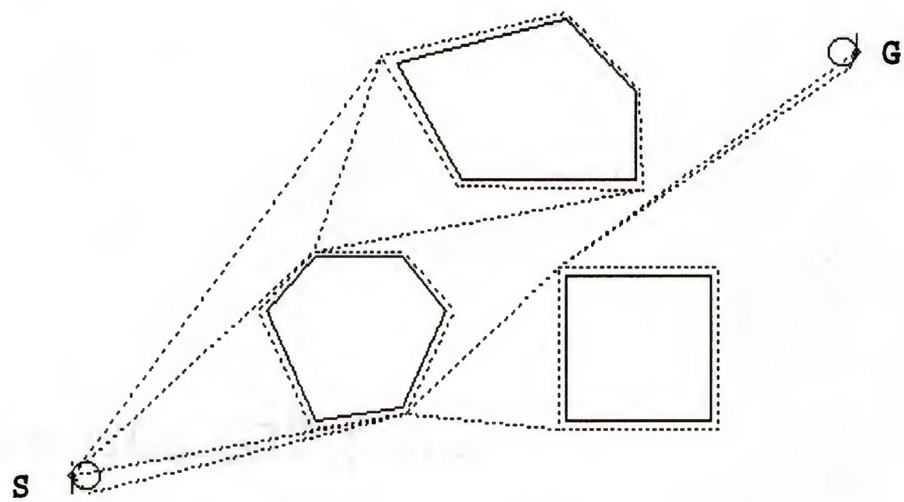


Figure 5.5.e Modification for Orientation

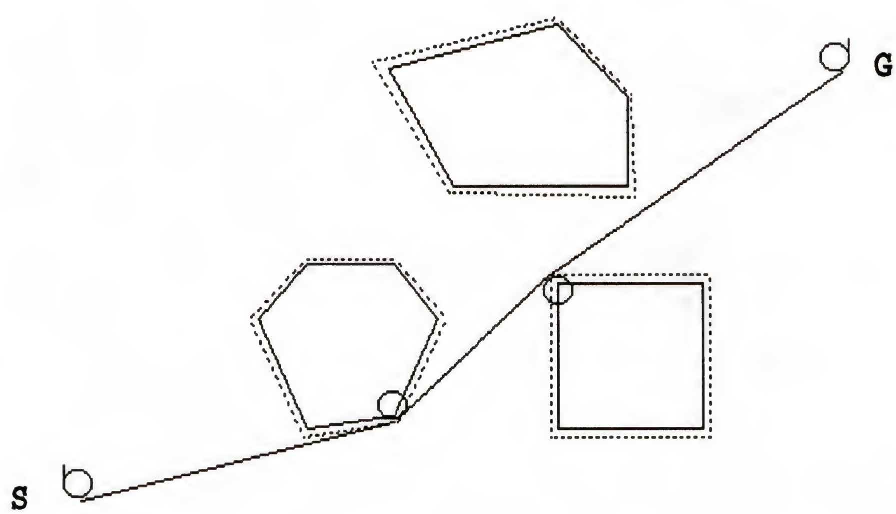


Figure 5.5.f Final Path

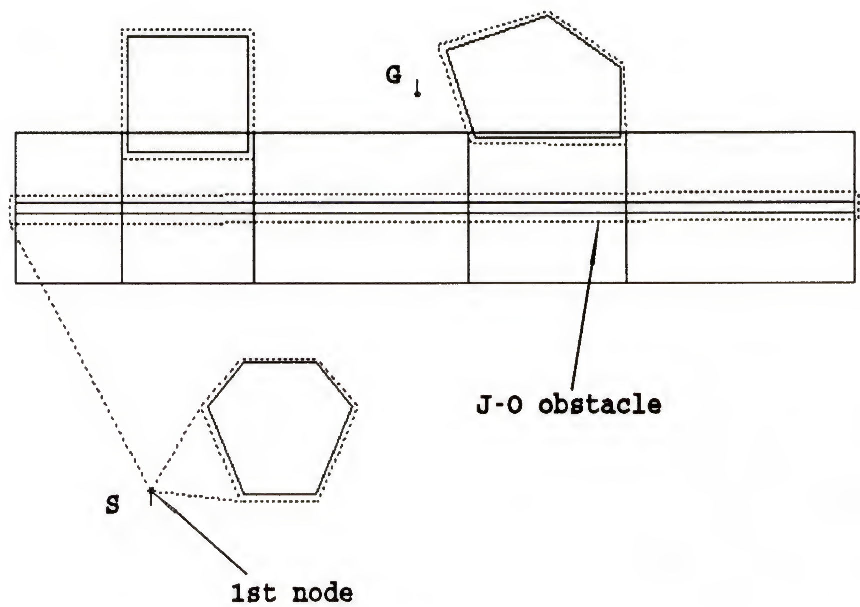


Figure 5.6.a Expansion of 1st Node

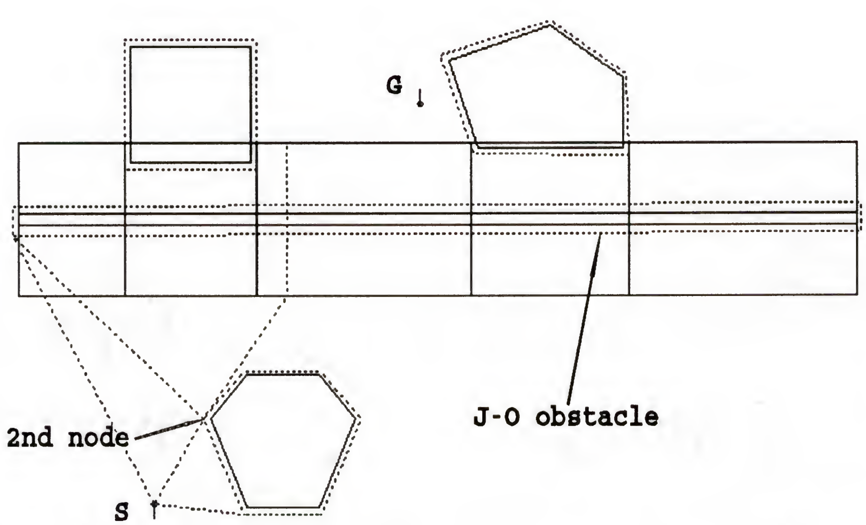


Figure 5.6.b Expansion of 2nd Node

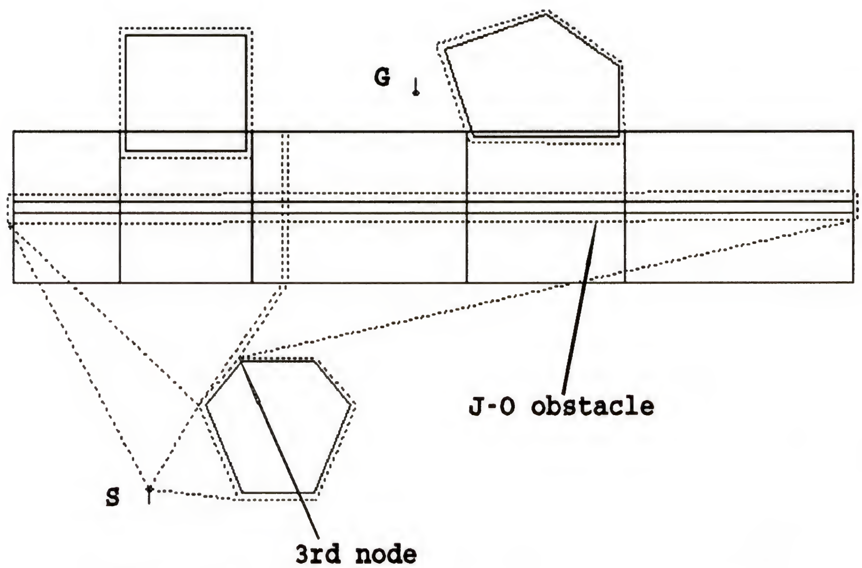


Figure 5.6.c Expansion of 3rd Node

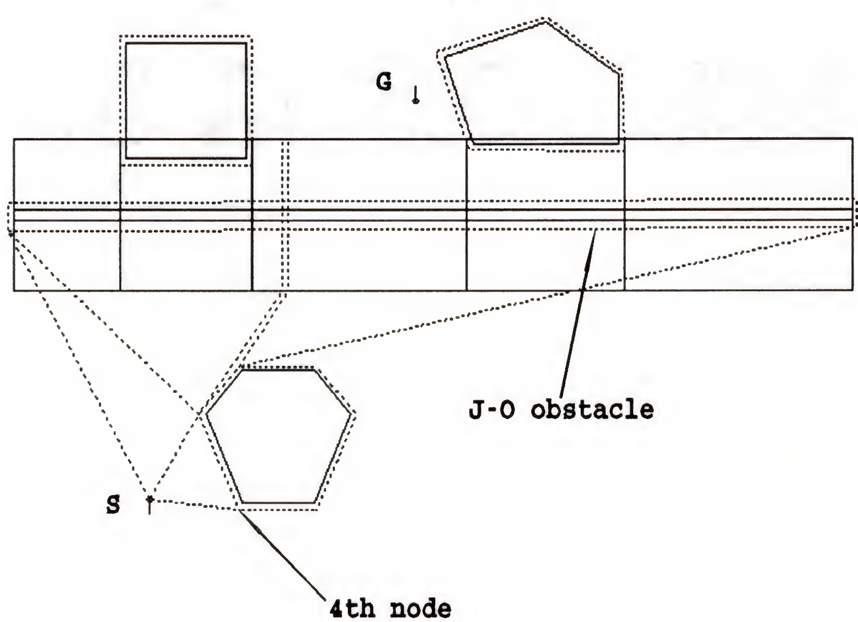


Figure 5.6.d Expansion of 4th Node

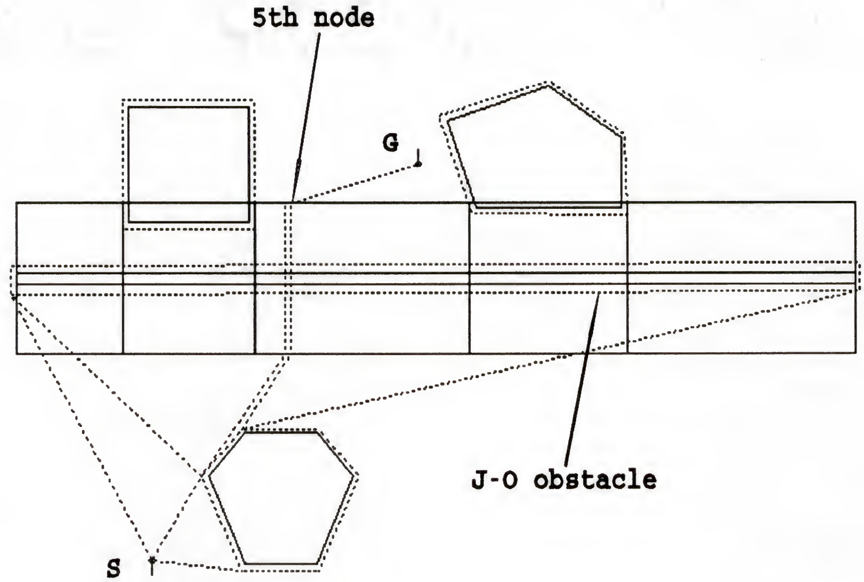


Figure 5.6.e Expansion of 5th Node

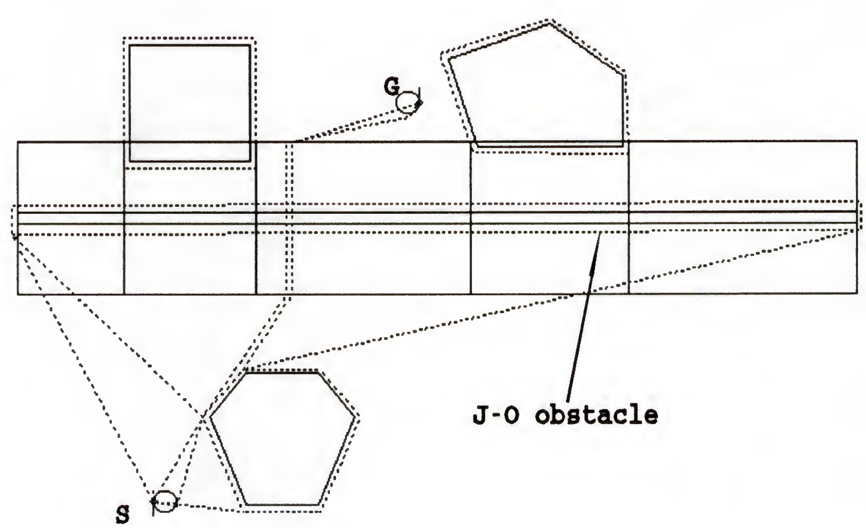


Figure 5.6.f Modification for Orientation

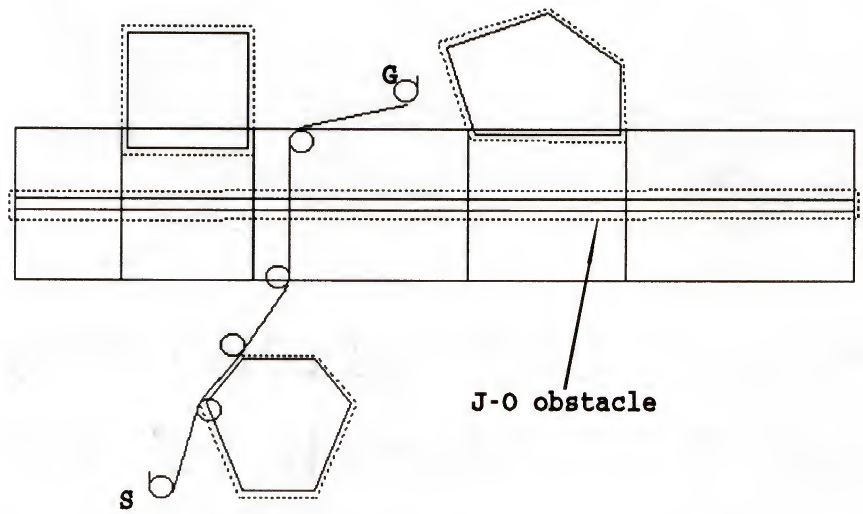


Figure 5.6.g Final Path

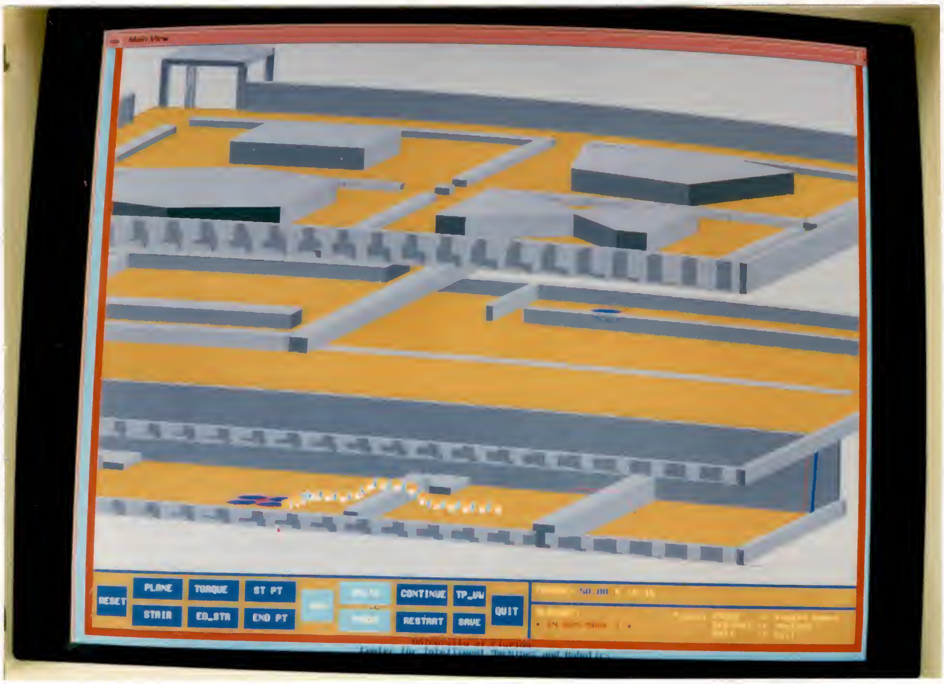


Figure 5.7 Vertical Path Over J-O Obstacle

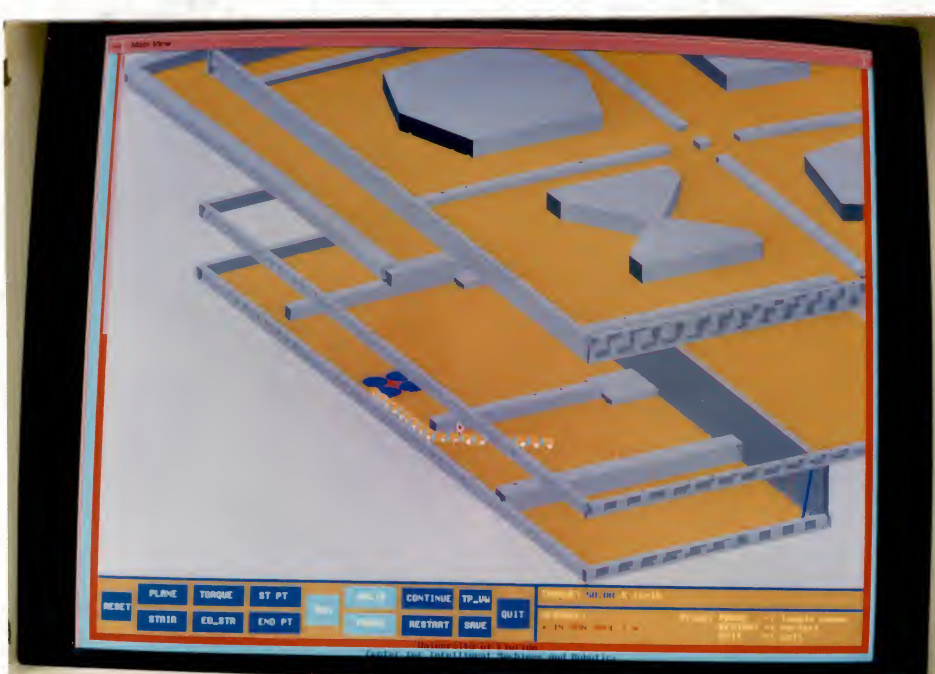


Figure 5.8 Horizontal Path Chosen to Pass J-O Obstacle

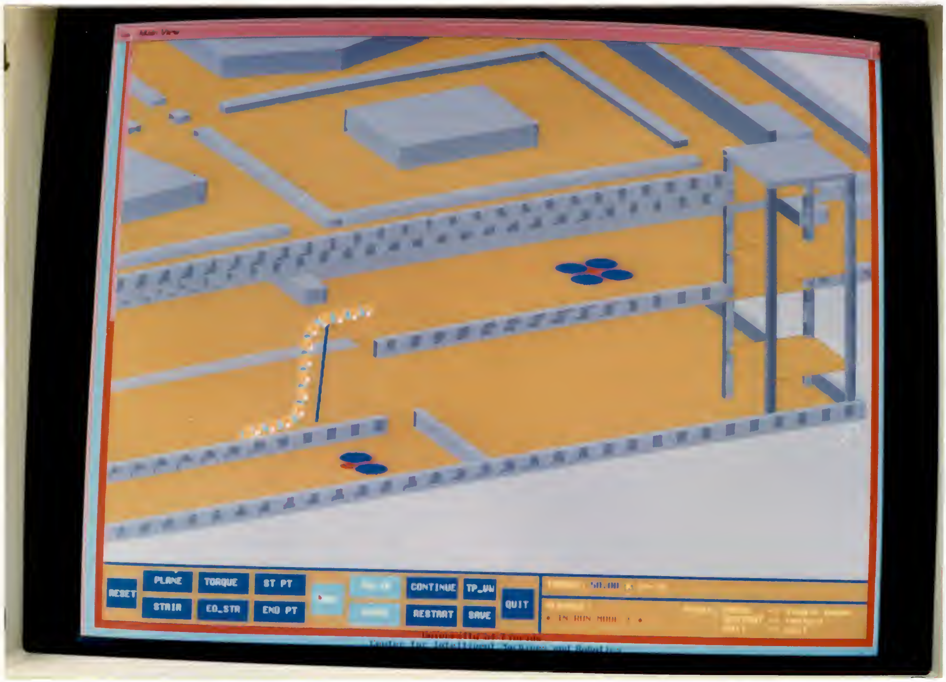


Figure 5.9 Vertical Path Between Planes
(Larger Joint Capacity)

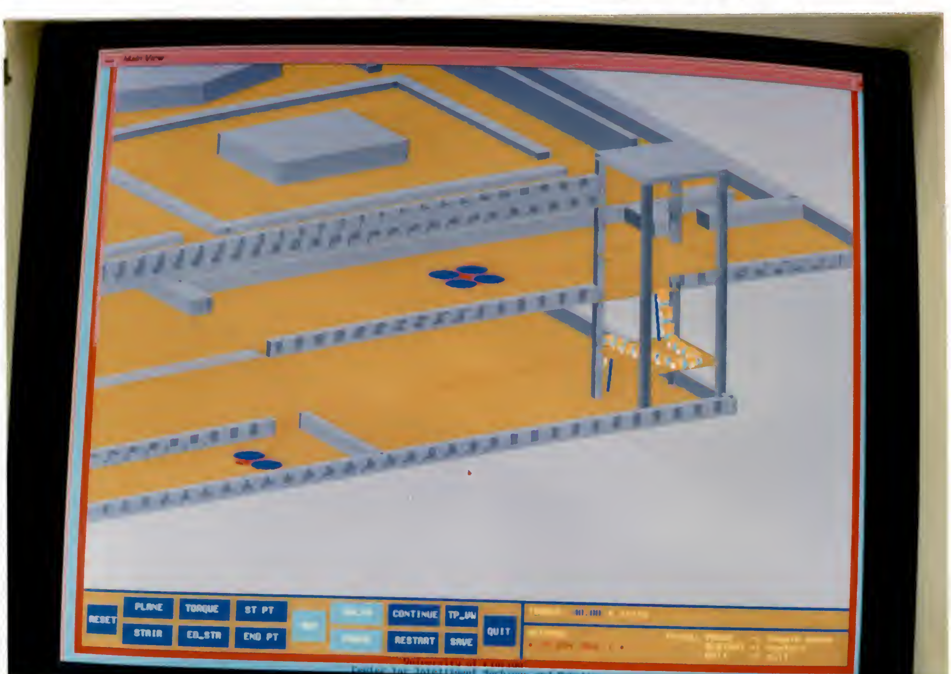


Figure 5.10 Vertical Path Between Planes
(Smaller Joint Capacity)

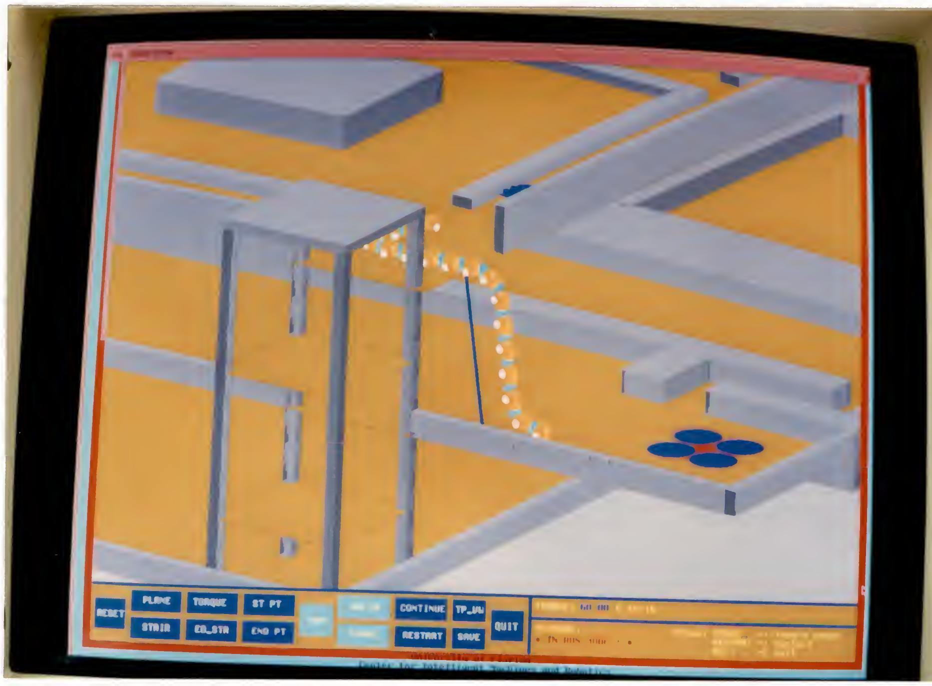


Figure 5.11 Vertical Path Between Planes
(Moves to a Lower Plane)

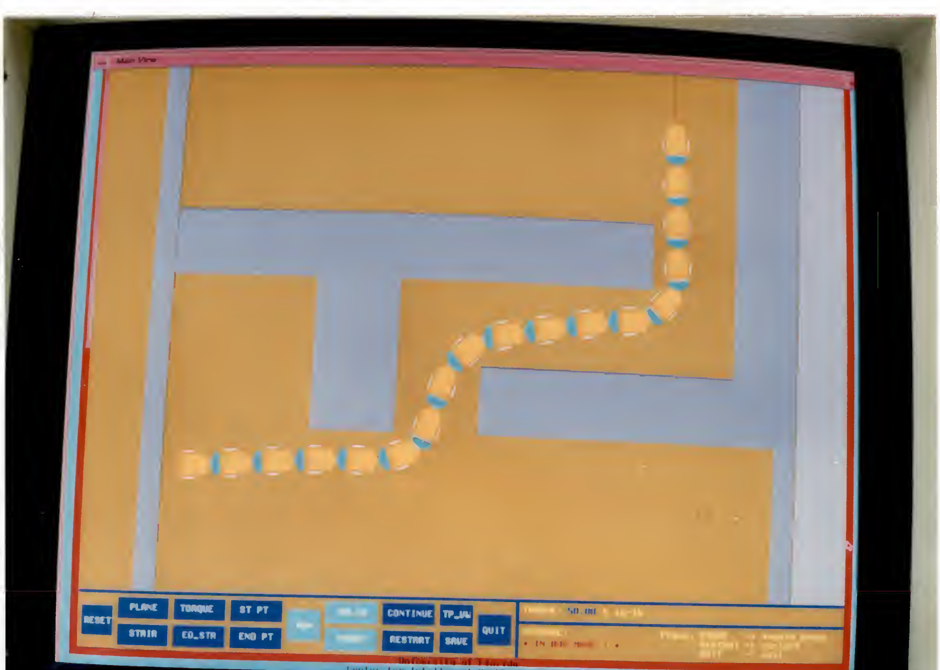


Figure 5.12 Passing Through the Labyrinth Entrance

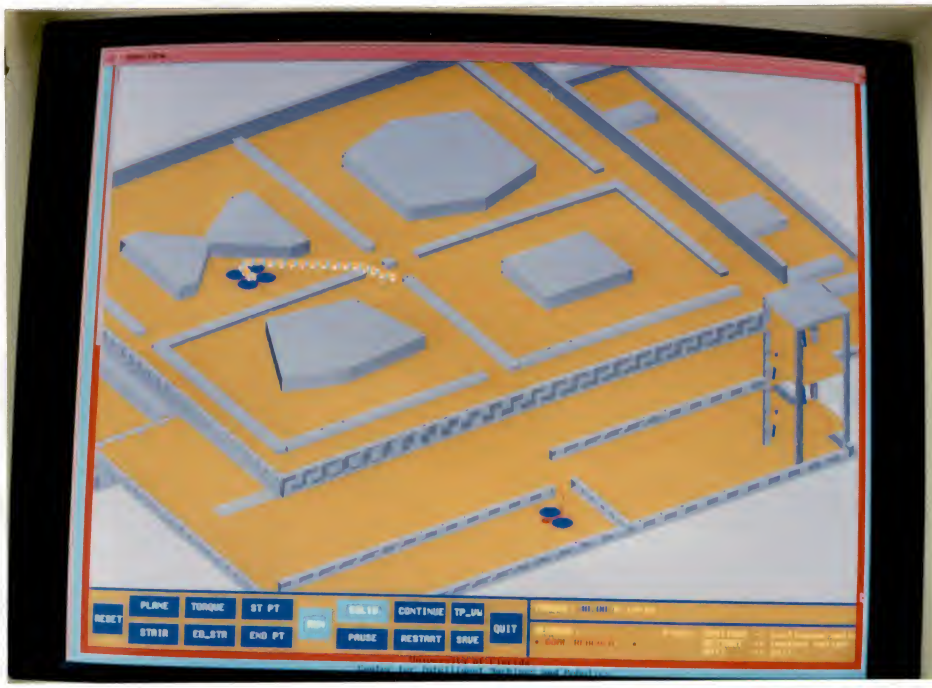


Figure 5.13 Horizontal Motion to the First Goal Point

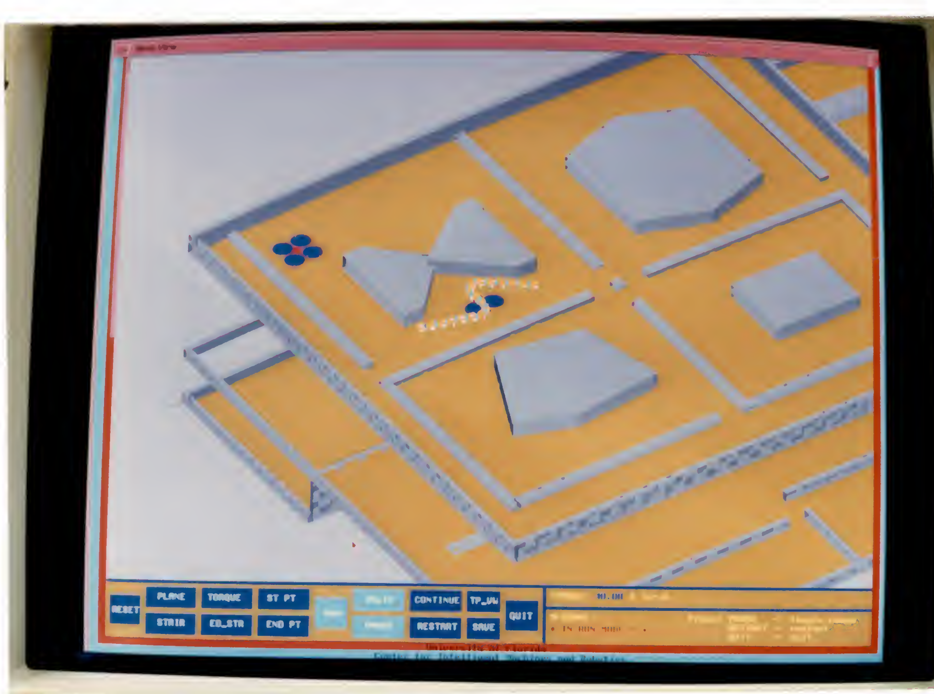


Figure 5.14 Continuous Movement to the Second Goal Point

CHAPTER 6
CONCLUSIONS AND RECOMMENDATIONS

In this research, a path planning strategy has been presented for planning a collision free path for the ATMS in a three-dimensional working environment. Algorithms which generate vertical and horizontal paths have been successfully developed and implemented. The vertical motion capability of the ATMS makes the path planning unique from others. In all other path planning problems, once the obstacle or free space has been defined, their characteristic is never changed in the process of path planning. However, in the path planning for the ATMS, an obstacle may not always be treated as an obstacle, and free space may not always be free space. "*An obstacle, not an obstacle; free space, not free space*" is the uniqueness of the path planning for the ATMS.

The representation of the environment of the ATMS is an important task in this research. A efficient representation of the environment saves tremendous efforts in the path planning process. Because of the requirement of a minimum turning radius for the ATMS, a non-uniform obstacle expansion method is introduced in this research. The method ensures that the nominal horizontal path which is planned through the

vertices of expanded obstacles can be easily modified into a collision free path for the ATMS to follow. A novel way of representing the J-O obstacle by buffer zones and pseudo obstacles makes the horizontal planning algorithm able to plan proposed vertical path and horizontal path segments utilizing the same procedures.

A hierarchical structure of path planning - vertical planning, horizontal planning, and operative planning, has been presented for planning a three-dimensional collision free path for the ATMS. In the vertical planning, by finding the via planes, the problem is broken down into path planning on each of the via planes. In the horizontal planning, a two-dimensional graph search (A^* search) with two designed graph building criteria plans the nominal horizontal path on each of the via planes. In the operative planning, due to the work in representing the environment, the nominal horizontal path is easily modified into a collision free path for the ATMS to follow.

The path planning algorithms have been successfully implemented on a Silicon Graphics 4D-310VGX workstation. Motions of the ATMS are animated in a designed working environment which consists of more than 40 obstacles to verify the results. The computation time of path planning varies from less than 1 second to approximately 3 seconds depending on the specified start and goal points.

In this research, there is only one access way between two horizontal planes. This actually limits a better path to be planned in the sense of overall path planning. The algorithm of horizontal planning can be extended to an environment which multiple access ways exist between two horizontal planes. In this case, all the sub-goal points on the horizontal plane will be considered in the graph search and choose the least cost point for the next expansion.

In this research, the collision avoidance is considered only between the ATMS and the obstacles in the environment. However, in the continuous movement of the ATMS from the first goal position to the second goal position, collision among the segments of the ATMS itself may happen. Though the collision can be avoided by a carefully chosen intermediate goal position, the avoidance of collision among the ATMS itself is still worthy of further investigation.

REFERENCES

[1] Ridgeway, S., Crane, C., Adist, P., and Harrell, R., "The Mechanical Design of a Parallel Actuated Joint for an Articulated Mobile Robot," Submitted to The 22nd ASME Mechanisms Conference, Phoenix, Arizona, September, 1992.

[2] Chiang, S. C., and Crane, C. D., "Algorithm Development and Computer Graphics Simulation of an Articulated Transporter/Manipulator System," Proc. of ASME International Computer in Engineering Conference, San Francisco, CA, 1988.

[3] Chiang, S. C., "Motion Analysis for an Articulated Transporter/Manipulator System," MS Thesis, Dept. Mech. Eng. University of Florida, 1988.

[4] White, J. R., "Surbot Makes Surveillance Safer," Nuclear Engineering International, April 1986, pp.36-41.

[5] White, J. R., Harvey, H. W., and Farnstrom, K. A., "Testing of Mobile Surveillance Robot at a Nuclear Power Plant," 1987 IEEE International Conference On Robotics & Automation, Vol. 2, pp. 714-719, 1987.

[6] Cruickshank, A. "REMOTEC's New Vehicle Uses the ANDROS Base," Nuclear Engineering International, April 1988, pp. 38-39.

[7] White, J. R., Coughlan, J. B., Harvey, M. W., Upton, R. G., and Walker, K. L., "Taking Andros for a Walk," Nuclear Engineering International, March 1989, pp. 52-53.

[8] Simmmons, R. K., "Smart Systems for Routine Jobs," Nuclear Engineering International, April 1986, pp. 41-42.

[9] Irving, T. L., "Surveyor Takes a Look Around Nine Mile Point," Nuclear Engineering International, April 1987, pp. 48-51.

[10] Weber, P. J., and Vanecek, C. W., "SIMON Combines Radiation Hardness with Computer Power," Nuclear Engineering International, May 1990, pp. 34-38.

[11] Weisbin, C. R., Burks, B. L., Einstein, J. R., Feezell, R. R., Manges, W. W., and Thompson, D. H., "HERMIES-III: A step toward autonomous mobility, manipulation and perception," Robotica, Vol 8, pp. 7-12, 1990.

[12] Naksno, Y., Fujie, M., Iwamoto, T., Kamejima, K., Sugiyama, K., and Funabashi, M., "A Concept for an Autonomous Mobile Robot and a Prototype Model," Proc. of '85 International Conference On Advanced Robotics, pp.413-420, 1985.

[13] Iwamoto, T., and Yamamoto, H., "Stairway Travel of a Mobile Robot with Terrain-Adaptable Craw Mechanism," Journal of Robotic System, Vol. 2(1), pp. 125-134, 1985.

[14] Obama, M., Ozaki, F., and Asano, K., "A Locomotive Inspection Robot for Turbine Building Interior Inspection in Nuclear Power Plants," Proc. of '85 International Conference On Advanced Robotics, Tokyo, Japan, pp. 355-362, 1985.

[15] Bartholet, T. G., "Technology for Mobil Robotics in Nonmanufacturing Application," Teleoperated Robotics in Hostile Environments, First Edition, RI/SME, Michigan, pp. 230-239, 1985.

[16] Carlton, R. E. and Bartholet, S. J., "The Evolution of the Application of Mobile Robotics to Nuclear Facility Operations and Maintenance," 1987 IEEE International Conference On Robotics & Automation, Vol.2, pp. 720-726, 1987.

[17] Köhler, G. W., Selig, M., and Salaske, M., "Manipulator Vehicles of the Nuclear Emergency Brigade in the Federal Republic of Germany," TELEOPERATED ROBOTICS IN HOSTILE ENVIRONMENTS, First Edition, RI/SME, pp. 197-219, 1985.

[18] Maeda, Y., Tsutani, S., and Hagihara, S., "Prototype of Multifunctional Robot Vehicle," Proc. of '85 International Conference On Advanced Robotics, Tokyo, Japan, pp. 421-428, 1985.

[19] Ohmichi, T., Hosaka, S., Nishihara, M., Ibe, T., Okino, A., Nakayama, J., Miida, T., and Ishida, M., "Development of the Multi Function Robot for the Containment Vessel of the Nuclear Plant," Proc. of '85

International Conference On Advanced Robotics, Tokyo, Japan, pp. 371-378, 1985.

[20] Hosaka, S., and Ohmichi, T., "Teleoperation of Robot for Maintenance and Inspection in the Containment Vessel of Nuclear Power Plant," Proc. of '85 International Conference On Advanced Robotics, Tokyo, Japan, pp. 379-386, 1985.

[21] Hirose, S., and Morishima, A., "Design and Control of a Mobile Robot with an Articulated Body," The International Journal of Robotics Research, Vol.9, No. 2, pp. 99-114, April, 1990.

[22] Lozano-Perez, T., and Wesley, M. A., "An Algorithm for Planning Collision-Free Paths Among Obstacles," Commun. ACM, vol.22, pp. 560-570, 1979.

[23] Takahashi, O., and Schilling, R. J.,"Motion Planning in a Plane Using Generalized Voronoi Diagrams," IEEE Trans. RA, Vol.5, No.2, pp. 143-150, 1989.

[24] Singh, J. S., and Wagh, M. D., "Robot Path Planning using Intersecting Convex Shapes," IEEE International Conference on Robotics and Automation, Vol.3, pp. 1743-1748, 1986.

[25] Udupa, S. M., "Collision Detection and Avoidance in Computer Controller Manipulators," Fifth International Joint Conference on Artificial Intelligence, MIT, Cambridge, Massachusets, pp. 737-748, 1977.

[26] Lozano-Perez, T., "Automatic Planning of Manipulator Transfer Movements," IEEE Trans. Sys., Man, Cybern., Vol. SMC-11, pp. 681-689, October, 1981.

[27] Lozano-Perez, T., "Spatial Planning: A Configuration Space Approach," IEEE Trans. Comp., Vol. C-32, pp.108-120, Feb. 1983.

[28] Andresen, F. P., Davis, L. S., Eastman, R. D., and Kambhampati, S., "Visual Algorithms for Autonomous Navigation," Proc. IEEE International Conference On Robotics And Automation, St. Louis, MO. pp. 856-861, 1985.

[29] Noborio, H., Naniwa T., and Arimoto, S., "A Quadtree-Base Path-Planning Algorithm for a Mobile Robot," Journal of Robotic Systems, Vol. 7(4), pp. 555-574, 1990.

[30] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robot," IEEE International Conference on Robotics and Automation, St. Louis, MO, pp. 500-505, March 25-28, 1985.

[31] Hague, T., Brady, M. and Cameron, S., "Using Moments to Plan Paths for the Oxford AGV," IEEE International Conference On Robotics And Automation, Vol.1, pp. 210-215, 1990.

[32] Kuan, D. T., Zamiska, J. C., and Brooks, R. A., "Natural Decomposition of Free Space for Path Planning," IEEE International Conference on Robotics and Automation, St. Louis, MO, pp. 168-173, 1985.

[33] Nilsson N.J., Principles of Artificial Intelligence, Tioga, Palo Alto, CA, pp. 74-84, 1980.

[34] Arkin, R. C., "Navigational path Planning for a Vision-Based Mobile Robot," Robotica, Vol.7, pp.49-63, 1989.

[35] Borenstein, J., and Koren, Y., "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," IEEE International Conference on Robotics and Automation, Vol. 1, pp. 572-577, 1990.

BIOGRAPHICAL SKETCH

Shih-chien Chiang was born in Tainan, Taiwan, R.O.C., on February 26, 1954. He received his Bachelor of Science degree in naval architecture and marine engineering from the National Cheng Kung University, Taiwan, in 1976. After two years' military service, he worked as a marine engineer at Tung Hwa Industry Co., Ltd., Taiwan. From 1979 to 1986, he worked as a mechanical engineer at King Fahad Hofuf Hospital in Saudi Arabia. He began his graduate studies in mechanical engineering at the University of Florida in August, 1986, and received the Master of Science degree in August, 1988.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Duffy, Chairman
Graduate Research Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Carl D Crane III, Cochairman
Assistant Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ali Seireg
Ebaugh Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

John Schueller
Associate Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Roy Harrell
Associate Professor of
Agricultural Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1992

for Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School